Foundation Models at Work: Fine-Tuning for Fairness in Algorithmic Hiring

Buse Sibel Korkmaz[1*], **Rahul Nair**[2], **Elizabeth M. Daly**[2], **Evangelos Anagnostopoulos**[3], **Christos Varytimidis**[3], **Antonio del Rio Chanona**[1]

[1]Imperial College London
[2]IBM Research Europe
[3]Workable

buse.korkmaz18@imperial.ac.uk, rahul.nair@ie.ibm.com, elizabeth.daly@ie.ibm.com, anagnostopoulos@workable.com, varytimidis@workable.com, a.del-rio-chanona@imperial.ac.uk

Abstract

Foundation models require fine-tuning to ensure their generative outputs align with intended results for specific tasks. Automating this fine-tuning process is challenging, as it typically needs human feedback that can be expensive to acquire. We present *AutoRefine*, a method that leverages reinforcement learning for targeted fine-tuning, utilizing direct feedback from measurable performance improvements in specific downstream tasks. We demonstrate the method for a problem arising in algorithmic hiring platforms where linguistic biases influence a recommendation system. In this setting, a generative model seeks to rewrite given job specifications to receive more diverse candidate matches from a recommendation engine which matches jobs to candidates. Our model detects and regulates biases in job descriptions to meet diversity and fairness criteria. The experiments on a public hiring dataset and a real-world hiring platform showcase how large language models can assist in identifying and mitigation biases in the real world. We open-source our proposed method and related resources [1].

Introduction

Foundation models have demonstrated exceptional capabilities in generating coherent and contextually relevant text (Radford et al. 2019; Brown et al. 2020; Taylor et al. 2022; Thoppilan et al. 2022; Touvron et al. 2023; Team 2023; Jiang et al. 2024). Large language models (LLMs) have accelerated progress in several natural language processing tasks, including text generation, translation, and sentiment analysis, among others (Liu, Shin, and Burns 2021; Sallam 2023; Lyu, Xu, and Wang 2023).

In practice, LLMs need to be adapted to specific tasks typically using a fine-tuning step. Fine-tuning aims to better align generative outputs on a specific task with desired outcomes. As a result, alignment research has emerged as a strategy that ensures the development of advanced AI systems resonates with intended goals and human values (Christiano et al. 2017; Yuan et al. 2023).

One prominent approach here is Reinforcement Learning from Human Feedback (RLHF). By leveraging human demonstrations, preferences, or feedback, RLHF guides the fine-tuning process, allowing them to learn and approximate human values (Stiennon et al. 2020). This method bridges the gap between human values and AI system behaviour, fostering a more robust and aligned decision-making process (Korbak et al. 2023). However, RLHF is resource-intensive, requiring extensive human annotations.

In this paper, we present a strategy where fine-tuning (alignment) is driven by the downstream task directly, i.e. *without human feedback*. We study this in the context of job description generation in hiring platforms where open jobs are matched to candidates using a recommendation system. We are interested in descriptions that appeal to a broad pool of candidates and do not marginalize specific groups.

Studies have shown that job postings using gender-neutral language have attracted a wider range of applicants than those with gender-biased terms (Woods, Tharakan, and Brown 2021). Moreover, seemingly innocuous phrases in job descriptions can deter potential candidates, especially those from underrepresented groups (Woods, Tharakan, and Brown 2021). For example, descriptions seeking "young and energetic" candidates can dissuade older individuals and suggest a lack of flexibility for those with other commitments.

In our setting, the risk of LLMs reinforcing societal stereotypes and prejudices is pronounced. LLMs can inadvertently inherit biases present in the underlying corpus (Bender et al. 2021). These biases can perpetuate unfairness, reinforce stereotypes, and marginalize certain social groups. For instance, language models trained on internet text data tend to exhibit gender and racial biases, leading to biased outputs when generating text or making predictions (Bolukbasi et al. 2016; Barikeri et al. 2021). Recognizing and tackling these biases is essential to ensure fairness in various downstream tasks.

Human preferences in this setting can be challenging to obtain from annotators for several reasons. Linguistic preferences are shaped by lived experiences that are varied (Davani et al. 2024). There is an absence of normative descriptions that can be objectively judged. A job description that appeals to Alice may not appeal to Bob. Crucially, it is difficult for humans to reason about the likely impacts of their preferences when generative outputs are used in broader al-

[*] This work was done during an internship at IBM Research.

[1] https://github.com/buseskorkmaz/FMs-at-work

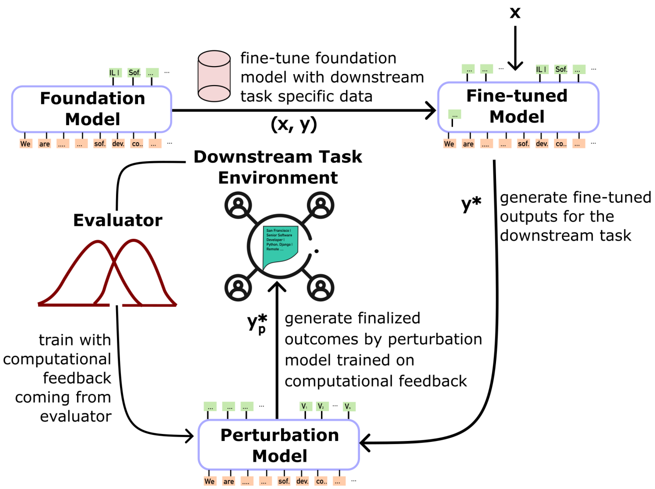
x
fine-tune foundation model with downstream task specific data
Foundation Model
(x, y)
Fine-tuned Model
IL l
Sof.
We
are
sof.
dev.
co..
Downstream Task Environment
San Francisco | Senior Software Developer | Python, Django | Remote ....
Evaluator
y*
generate fine-tuned outputs for the downstream task
y_p^*
generate finalized outcomes by perturbation model trained on computational feedback
train with computational feedback coming from evaluator
Perturbation Model
V_1
V_2
V_n

Figure 1: Our methodology *AutoRefine* works by building a perturbation model that assesses the alignment of the generated content with task-specific goals. Evaluations serve as computational feedback that iteratively updates the perturbation model. During generation, both the original and perturbation models are used to generate tokens.

gorithmic settings.

The objective of this work is to propose a framework for fine-tuning foundation models without human feedback, by quantifying their impact on downstream tasks. In our study, we examine the influence of linguistic biases on a recommendation system, taking job description generation as a concrete example. The diversity of candidates matched to these generated descriptions is a primary concern, emphasizing the need for fairness and bias mitigation.

Our primary contributions are on: *(a) task-responsive fine-tuning:* We propose a novel approach for fine-tuning foundation models using reward signals derived from measurable outcomes in downstream tasks. This method facilitates precise model adjustments based on actual task impacts, bypassing traditional reliance on human feedback collection or preference modelling, *(b) bias mitigation for foundation models:* We demonstrate our method for bias mitigation and fairness when downstream tasks need to satisfy established equity criteria. As biases are quantified and mitigated during fine-tuning, our method ensures that generated content is purpose-aligned and inclusive, and *(c) application to job description generation:* We present a real-world use case in job description generation, showcasing the practical challenges and our solutions to ensure that attract a diverse and fair candidate pool.

Related work is relegated to Appendix. We describe our method next.

Method

We propose *AutoRefine*, a method for fine-tuning foundation models without requiring human feedback. The process works in three stages: first aligning the model to a specific downstream task, then introducing a perturbation mechanism to optimize outputs based on performance metrics, and finally deploying the combined system. Below, we present our problem formulation and explain in detail how *AutoRefine* reduces bias in job descriptions using feedback from a recommender system.

Problem formulation

Our framework consists of three core components: a pretrained foundation model $\mathcal{M}$, a specific downstream task $\mathcal{T}$, and a performance evaluator E that measures how well $\mathcal{M}$'s outputs perform. Our goal is to fine-tune $\mathcal{M}$ so its outputs for task $\mathcal{T}$ improve based on feedback from E. We now describe each component and our training process in detail.

Downstream task environment. The downstream task environment represents the specific task $\mathcal{T}$ for which the foundation model $\mathcal{M}$ is being fine-tuned and optimized on. It takes the generated fine-tuned outputs y^* from the fine-tuned model $\mathcal{M}^*$ as input and interacts with the evaluator E to assess the performance of the generated outputs.

The downstream task environment can vary depending on the specific application. For example, in our context of job description generation, the downstream task environment involves matching job descriptions with candidates. The task environment provides the necessary context and data for evaluating the generated outputs against the desired task-specific goals.

Fine-tuned model. The first step in our training process is to fine-tune the foundation model $\mathcal{M}$ using a supervised dataset $\mathcal{D}$ consisting of prompts x and their corresponding expected responses y. This step is not fairness-aware and aims to fine-tune the model $\mathcal{M}$ to produce similar responses to y for a given prompt x. The objective is to minimize:

$$\mathcal{L}(\mathcal{M}) = -\mathbb{E}_{(x,y)\sim\mathcal{D}}[\log P(y|\mathcal{M}(x))]. \tag{1}$$

Here, $P(y|\mathcal{M}(x))$ is the likelihood of $\mathcal{M}$ producing y for prompt x. We minimize $\mathcal{L}$, ensuring the model's output aligns with the dataset. The fine-tuned model is defined as:

$$\mathcal{M}^* = \arg\min_{\mathcal{M}} \mathcal{L}(\mathcal{M}) \tag{2}$$

where $\mathcal{M}^*$ produces y^* for x.

Evaluator. The evaluator E serves as a computational feedback mechanism that assesses the performance of the generated outputs from the perturbation model $\mathcal{M}_p^*$ in the context of the downstream task environment. It takes the generated outputs y^* from the downstream task environment $\mathcal{T}$ and evaluates them based on predefined metrics. The evaluator provides a quantitative measure of how well the generated outputs align with the desired task-specific goals by assigning rewards or scores to the generated outputs, which serve as feedback signals to guide the training of the perturbation model.

In the context of fairness of job descriptions, the evaluator assesses the difference between the targeted and realized diversity metrics. The evaluator assigns higher rewards to job descriptions that meet the desired fairness criteria and lower rewards to those that lack diversity. The specific implementation of the evaluator may vary depending on the downstream task and the desired optimization objectives.

The rewards or scores generated by the evaluator are then used to update the perturbation model $\mathcal{M}_p$ through a reinforcement learning approach. The perturbation model learns to adjust the output probabilities of the fine-tuned model based on the feedback received from the evaluator. This iterative process allows the methodology to optimize the generated outputs towards the desired task-specific goals.

Perturbation model. To train $\mathcal{M}_p$, we employ a reinforcement learning (RL) approach. The perturbation model learns a value function $V(y^*)$ that estimates the expected future reward for generating output y^*. The value function is learned through interactions with the fine-tuned model $\mathcal{M}^*$ and the downstream task evaluator $\mathcal{T}$.

The perturbation model $\mathcal{M}_p$ is defined as a function that modifies the output probabilities of $\mathcal{M}^*$ based on the learned value function:

$$\mathcal{M}_p(y^*|x) \propto \mathcal{M}^*(y^*|x) f(\beta, V(y^*)) \tag{3}$$

where $f(\beta, V(y^*))$ is a function that takes the temperature parameter β and the value function $V(y^*)$ as inputs and returns a non-negative value that scales the probabilities of the fine-tuned model $\mathcal{M}^*$. In our implementation, we choose the exponential function $f(\beta, V(y^*)) = e^{\beta V(y^*)}$ for its desirable properties and simplicity.

The training objective for $\mathcal{M}_p$ is to maximize the expected reward while maintaining proximity to the fine-tuned model's output distribution:

$$\mathcal{L}(\mathcal{M}_p) = \mathbb{E}_{y^* \sim \mathcal{M}^*(x)}[E(\mathcal{T}(\mathcal{M}_p(y^*)))]. \tag{4}$$

The training process involves iteratively generating outputs from the fine-tuned model $\mathcal{M}^*$, applying the perturbation model $\mathcal{M}_p$ to obtain perturbed outputs, evaluating the perturbed outputs using E, and updating the value function and perturbation model based on the received rewards. During inference, the optimized perturbation model $\mathcal{M}_p^*$ regulates the outputs of the fine-tuned model $\mathcal{M}^*$ to generate better-aligned outputs with the downstream task goals.

Reinforcement learning for fine-tuning

In *AutoRefine*, we employ RL to align the fine-tuned foundation model for a downstream task, similar to previous approaches (Bai et al. 2022; Ouyang et al. 2022; Askell et al. 2021). Our choice is motivated by RL's inherent capability to handle dynamic feedback and optimize non-differentiable objectives, aligning well with the performance evaluator E in our problem formulation. RL's strength in managing the exploration-exploitation trade-off ensures that our model generates diverse yet contextually relevant outputs. Given the sequential nature of text generation, we can model the prompt as a state and each generated token as an action and quantify the reward as feedback from the evaluator E for choosing a particular token given the state, without the need for explicit human annotations. This feedback shapes the reward function within the RL step of *AutoRefine*, and any RL algorithm adept at managing such non-differentiable feedback can be employed in this stage.

We use the Implicit-Language Q-Learning (ILQL) algorithm (Snell et al. 2023) to train the perturbation model. ILQL is an offline reinforcement learning algorithm tailored specifically for language models. The primary advantages of ILQL in our context are twofold. Firstly, ILQL learns from a token-level Q-function which allows us to *identify bias at the token level*. This representation offers a tangible metric that can be strategically optimized to manage and mitigate bias. Secondly, as being based on offline learning, it leverages samples from an existing dataset and *reduces the queries to a recommendation engine* during training. Quantifying the reward for the generated text through the recommendation engine is a bottleneck in our setting due to the large size of the candidate pool such as a million candidates in the real-hiring platform data.

In this step of *AutoRefine*, the agent, which in our context is the language model, interacts with an environment to produce sequences, such as job descriptions. Feedback, in the form of rewards, is provided based on the inherent bias of the sequences produced. Q-value formulates the anticipated cumulative reward for a specific sequence. By leveraging this Q-value, the agent is trained to produce sequences that strike a balance between *high quality and minimal bias*, similar to loss function-based debiasing techniques (Barikeri et al. 2021).

Debiasing Job Descriptions

Our goal is to create job descriptions that attract a broader pool of qualified applicants. In practice, algorithmic hiring tools recommend candidates for job openings through recommendation engines. For our research, we develop a proxy recommendation engine to serve as our evaluator E, simulating how such systems would operate in real-world recruitment platforms. We use *AutoRefine* to generate job descriptions that achieve both effectiveness and inclusivity, refining the foundation model $\mathcal{M}$ based on feedback from this performance evaluator E.

The process begins with supervised fine-tuning (Equation 1), which teaches the foundation model to generate appropriate job descriptions while maintaining its ability to produce coherent, relevant content. We then enhance these outputs by introducing carefully calibrated perturbations through $\mathcal{M}^*$, with the specific aim of improving diversity outcomes. The objective function for $\mathcal{M}_p$, detailed in Equation 3, guides the model to make adjustments that better align the generated content with our diversity criteria. When deployed, this integrated system produces job descriptions that are both relevant to the position and meet established diversity standards.

For implementation, we selected GPT-2 as our base model and fine-tuned it on job description data, enabling it to generate appropriate descriptions from detailed job specifications. To address potential bias in the generated content, we then apply the second phase of *AutoRefine*: training a perturbation model that incorporates feedback from the downstream task using ILQL (Snell et al. 2023). During actual use, the system takes an original job description as an input. The perturbation model then generates an improved, less biased version by re-ranking the fine-tuned model's token outputs based on maximum potential reward. The following sections detail our reward function and explain how we optimize the learning process to effectively regulate application bias.

Downstream task environment: Job-candidate matching engine

To evaluate our approach, we develop a proxy recommendation system that simulates how algorithmic hiring tools match candidates to job openings. While bias in algorithmic hiring can originate from multiple sources, we specifically focus on bias stemming from job posting language rather than potential biases within recommendation systems themselves. This focus is important because job posting language has broader implications beyond algorithmic systems - it shapes how candidates perceive both the role and the company (Woods, Tharakan, and Brown 2021). Our proxy system simulates both human application decisions and algorithmic hiring recommendations.

Our evaluation process has two main phases. First, the perturbation model modifies the fine-tuned LLM's token logits to reduce bias in the job posting. Then, we evaluate the rewritten description using our recommendation engine. This evaluation begins by filtering candidates based on the position's hard requirements. We then use BERT (Devlin et al. 2019) to generate embeddings for both the job descriptions and candidate profiles. By computing cosine similarity between these embeddings, we identify the top k candidates most similar to the job description. These candidates are then analyzed by our fairness evaluator to compute a diversity score.

Fairness evaluator: Reward function driven by diversity

Our approach evaluates fairness by comparing candidate distributions across two dimensions: gender and geolocation. We analyze these by measuring the difference between two probability distributions - the realized distribution D_{realized} from our selected candidates and a target distribution D_{target}. To quantify this difference, we employ the 1-Wasserstein distance, which provides values between 0 and 1. Here, 0 indicates perfectly matching distributions, while 1 represents complete divergence (where one distribution is concentrated at 0 and the other at 1).

For gender analysis, we compute the 1-Wasserstein distance between the actual gender distribution of candidates matched to a rewritten job description y_p^*, denoted as $D_{\text{realized, gender}}(y_p^*)$, and our target distribution $D_{\text{target, gender}}$:

$$\Delta_{\text{gender}}(y_p^*) = W_1(D_{\text{realized, gender}}(y_p^*), D_{\text{target, gender}}). \quad (5)$$

The geolocation attribute $\Delta_{\text{geolocation}}(y_p^*)$ can be computed similarly to the 1-Wasserstein distance between the realized and the target distributions.

These measurements combine to create our diversity score, which serves as the reward in our reinforcement learning environment. The score captures the total distribution mismatch across both attributes:

$$\mathcal{R}(y_p^*) = \Delta_{\text{gender}}(y_p^*) + \Delta_{\text{geolocation}}(y_p^*). \quad (6)$$

A smaller Wasserstein distance between the achieved and target distributions indicates a higher diversity score, implying that the job description is more aligned with our diversity goals.

Metrics

Diversity score. To evaluate our model's effectiveness, we apply *AutoRefine* to rewrite job descriptions, focusing on the roles shown in Figure 2. We compare diversity scores between original and rewritten descriptions. The effectiveness of our bias regulation is demonstrated when rewritten descriptions show smaller gaps between observed and target distributions compared to the original descriptions.

Impact ratio. We evaluate fairness using metrics established by New York Local Law 144 (New York City Department of Consumer and Worker Protection 2023), which provides a framework for auditing bias in algorithmic recruitment systems. The law introduces two key metrics to ensure transparency and equity: the *selection rate* (measuring a cohort's historical success in being selected) and the *impact ratio* (comparing a group's selection rate to that of the best-performing group). While the law covers multiple demographic categories including gender, race, ethnicity, and their intersections, our analysis focuses specifically on gender and location bias. We implement these metrics in our setting as follows: For each group g within a category $\mathcal{G}$ and for each job opening, we calculate selection rate using: (i) Selected candidates: The number of candidates from group g appearing in the top-10 recommendations, (ii) Candidate pool: All relevant candidates from group g, where relevance is determined by cosine similarity (taking the top 50 candidates).

The selection rate for each group is calculated as:

$$\text{SR}_g = \frac{\text{top-10 candidates from } g}{\text{\# of relevant candidates from } g} \qquad \forall g \in \mathcal{G}. \quad (7)$$

The impact ratio (IR) is measured as the impact ratio relative to the best-performing group.

$$\text{IR}_g = \frac{\text{SR}_g}{\max_{g' \in \mathcal{G}} \text{SR}_{g'}} \qquad \forall g \in \mathcal{G}. \quad (8)$$

The best-performing group has $\text{IR}_g = 1$. Values close to 1 indicate equity across groups for that categorization. Values further away from 1 indicate potential bias.

TPR-GAP (True Positive Rate GAP). We adopt the fairness metric of TPR-GAP introduced by De-Arteaga et al. (2019) to our context. TPR represents the fraction of relevant candidates from a specific group g that are included in the top-k recommendations. The relevancy of a recommended candidate is decided by whether the profession of the matched candidate is the same position in the job advertisements.

For each group g in a category $\mathcal{G}$, we define the True Precision Rate (TPR) as:

$$\text{TPR}_g = \frac{\text{relevant candidates from } g \text{ in top-}k}{\text{total relevant candidates from } g} \qquad \forall g \in \mathcal{G}. \quad (9)$$

The TPR-GAP measures the difference in TPR between the best-performing group and the worst-performing group within a category $\mathcal{G}$:

$$\text{TPR-GAP} = \max_{g \in \mathcal{G}} \text{TPR}_g - \min_{g \in \mathcal{G}} \text{TPR}_g. \tag{10}$$

A smaller TPR-GAP indicates higher fairness across groups, as it suggests that the top-k recommendations include a similar proportion of relevant candidates from each group. Conversely, a larger TPR-GAP indicates potential bias, as it implies that the job-candidate matching system favors certain groups over others in terms of including relevant candidates in the top-k recommendations.

Experiments

We conduct experiments on three datasets: the open-source dataset of Hackernews hiring posts and candidate profiles[2], Bias in Bios candidate profiles[3], and a large dataset of job specifications and candidates from a real-world hiring platform[4]. The datasets are described in Appendix.

Baselines

We compare the performance of our proposed approach, *AutoRefine*, with several debiasing algorithms. These algorithms represent different approaches to debiasing, ranging from embedding-level modifications to prompt-based techniques that we briefly introduce next. Ravfogel et al. (2020) proposed Iterative Null-space Projection (INLP), a method for debiasing embeddings by iteratively projecting them onto the null-space of protected attributes. This approach aims to remove information related to sensitive attributes from the embeddings while preserving their utility for downstream tasks. Liang et al. (2020) introduced Sentence-Level Debiasing (SentD in Table 1), an algorithm designed to debias pre-trained contextual embeddings at the sentence level, focusing on removing biases present in the representations of sentences and enabling more equitable downstream applications. Schick, Udupa, and Schütze (2021) developed Self-Debiasing (SD), a debiasing technique for GPT-2 models where the fine-tuned model self-diagnoses the bias present in the generated text and removes it, resulting in more neutral and unbiased outputs. Finally, Morabito, Kabbara, and Emami (2023) proposed Instructive-Debiasing (ID), an algorithm that utilizes debiasing prompts containing specific information about the category of bias present in a given text. By providing explicit instructions, the model learns to generate text that is less biased with respect to the specified categories.

Results

Fairness (Hacker News). To assess fairness, we compare the diversity scores and impact ratios between the original and rewritten job descriptions for all tested methods. Table 1 shows a 14% improvement in the *AutoRefine* rewritten descriptions compared to the original Hacker News posts. The reduced magnitude of the diversity score indicates a closer alignment with the desired diversity targets. Furthermore, the gender-specific impact ratios provide a more granular view of the alignment. For instance, the IR values for both male and female demographics remain consistent between the original and rewritten descriptions with a slight increase in IR_{male}. While our method is superior to other debiasing algorithms in terms of diversity score, the IR values of all methods are significantly close to each other. We also assess the IR focusing on geolocation, and the results do not exhibit significant differences among the methods. Nonetheless, we report these scores in the appendix for completeness.

Table 13 showcases specific examples from the evaluation set, highlighting the modifications made by the RL agent. The edits, though seemingly minor, have substantive implications for gender inclusivity and overall alignment with diversity goals. For instance, phrases that might be perceived as gender-biased or non-inclusive are either replaced or refined to ensure neutrality and inclusivity. Terms like "maniacally focused on" are changed to "dedicated to", and specific gendered or potentially exclusionary terms are redacted or replaced, ensuring the descriptions are more universally appealing. More examples are given in Table 14. A key observation from the modifications made by the RL agent and their subsequent influence on the impact ratio is the profound effect of subtle changes on the downstream task. These nuanced alterations, despite their seemingly minor nature, can have significant effects. This phenomenon further underscores the challenges faced by human evaluators during feedback collection. Such subtle changes are often not easy to catch by human evaluators, emphasizing the intricacies of the task at hand and the need to develop alternatives to human feedback for fine-tuning foundation models.

Fairness (Bias in Bios). As seen in Table 2, the rewritten ads with fairness considerations reduce the TPR-GAP for 4 out of 5 occupations (except for the accountant role), which shows an improvement over the original job descriptions. We calculated the GAP as $\text{TPR}_{\text{female},y}$ - $\text{TPR}_{\text{male},y}$ where y is all considered occupations.

Fairness (Hiring Platform Data). We fine-tune GPT-2 using our proposed algorithm using both location and gender as diversity measures to optimize. In evaluation, we consider the impact ratio statistic for various groups of interest. We omit intersectional analysis for brevity. Table 3 shows the impact ratio improving for female candidates. Job specification changes, however, had no impact on location-specific diversity in this instance.

In Table 1, we present the significant advantage of our algorithm. Since our training includes supervised fine-tuning specific to the domain and then cleansing generated text with more application-oriented bias, *AutoRefine* produces much less non-sensible response as a rewritten job description. To demonstrate that, we benefit UniEval (Zhong et al. 2022) language quality platform and evaluate language quality metrics in the aspects of naturalness, coherence, groundedness, and understandability for rewritten job descriptions

[2] https://huggingface.co/datasets/dansbecker/hackernews_hiring_posts

[3] https://huggingface.co/datasets/LabHC/bias_in_bios

[4] https://www.workable.com/

	Fairness			Text Quality			
Model	Diversity Score	IR$_{\text{female}}$	IR$_{\text{male}}$	Naturalness	Coherence	Groundedness	Understand.
Original	-23.48	0.84	0.76	0.57	1.0	1.0	0.64
GPT-2-large	↑7.4% -21.75	↑2.4% 0.86	↑1.3% 0.77	↓18% 0.47	↓3% 0.97	↓3% 0.97	↓17% 0.53
+INLP-race	↑7.0% -21.83	0.84	↑1.3% 0.77	↓70% 0.17	↓26% 0.74	↓70% 0.30	↓70% 0.19
+INLP-gender	↑7.0% -21.83	0.84	↑1.3% 0.77	↓70% 0.17	↓26% 0.74	↓69% 0.31	↓70% 0.19
+SentD-race	↑7.0% -21.83	0.84	↑1.3% 0.77	↓70% 0.17	↓26% 0.74	↓70% 0.30	↓70% 0.19
+SentD-gender	↑7.2% -21.78	0.84	↑2.6% 0.78	↓63% 0.21	↓78% 0.22	↓84% 0.16	↓64% 0.23
+SD	↑6.6% -21.92	0.84	↑2.6% 0.78	↓18% 0.47	↓3% 0.97	↓3% 0.97	↓17% 0.53
+ID	↑6.2% -22.02	↑2.4% 0.86	↓2.6% 0.74	↓37% 0.36	↓13% 0.87	↓16% 0.84	↓39% 0.39
+AutoRefine	↑14.1% **-20.17**	0.84	↑2.6% **0.78**	**0.57**	**1.0**	**1.0**	**0.64**

Table 1: Benchmarking of debiasing approaches comparing fairness and text quality metrics. Changes shown as percentages relative to original baseline. The highlighted (bold) scores show our method maintains original text quality while achieving the best improvement in diversity metrics.

	Original	AutoRefine
IR$_{\text{male}}$	0.89 ± 0.20	0.97 ± 0.10
IR$_{\text{female}}$	0.70 ± 0.26	0.55 ± 0.24
TPR-GAP$_{\text{software-eng}}$	0.024	0.005
TPR-GAP$_{\text{attorney}}$	0.009	-0.007
TPR-GAP$_{\text{accountant}}$	-0.006	0.023
TPR-GAP$_{\text{professor}}$	0.042	0.035
TPR-GAP$_{\text{journalist}}$	-0.038	-0.012
Diversity Sc.	-6.135	-5.93

Table 2: Key fairness measures on job rewriting experiments with Bias in Bios candidates dataset.

	Original	AutoRefine	p-value
IR$_{\text{female}}$	0.618±0.37	0.668±0.38	0.069*
IR$_{\text{male}}$	0.634±0.38	0.587±0.38	0.936
IR$_{\text{unknown}}$	0.621±0.35	0.607±0.35	0.750
IR$_{\text{africa}}$	0.250±0.40	0.181±0.36	0.995
IR$_{\text{asia}}$	0.438±0.40	0.462±0.39	0.212
IR$_{\text{eu}}$	0.364±0.39	0.322±0.39	0.995
IR$_{\text{na}}$	0.563±0.31	0.586±0.32	0.522
IR$_{\text{oceania}}$	0.252±0.42	0.265±0.43	0.334
IR$_{\text{sa}}$	0.063±0.24	0.030±0.16	0.990
Diversity Sc.	-20.96±9.83	-22.67±9.51	0.961

Table 3: Key fairness measures (mean ± standard deviation) on job rewriting experiments on Hiring Platform data before and after re-writes. Higher values are better. p-values from a binomial test showing the impact of rewrites (* implies significance at 10%).

of each algorithm in our benchmarking suite. Where it is required, we include the original job ad as a reference text in the language quality evaluation.

Our benchmarking results show that the proposed approach of implementing minimal token-based changes does not hurt the language quality while fairness metrics in Table 1 demonstrate our algorithm is competitive from the debiasing perspective. Moreover, the substantially worse language quality of the existing debiasing algorithms on text quality evaluation suggests the importance of reporting text quality-based scores for debiasing methods which alter language generation mechanics of underlying pre-trained methods and may degrade the quality of outputs.

Impact of fairness on recommendation quality

We investigate the potential impact of rewritten job advertisements on the utility of matching with the best and most qualified candidates. We evaluate the quality of the recommendations using metrics commonly employed in the recommender systems literature. Specifically, we consider Mean Reciprocal Rank (MRR) and Normalized Discounted Cumulative Gain (NDCG) to assess the effectiveness of matching job descriptions with candidate profiles.

Table 4 presents the comparison of these metrics for the original job descriptions and our generated job descriptions across different professions at various top-k values (10, 25, and 50). The top-k values represent the number of top-ranked candidates considered for each job description. The results indicate that there is no substantial impact on the utility of the job descriptions after the rewriting process. In most cases, our generated job descriptions exhibit slightly higher match qualities compared to the original descriptions, as evident from the marginally higher values of MRR and NDCG across different top-k values.

For instance, considering the profession of accountant at top-10, the MRR@10 values for the original and generated descriptions are 0.614 and 0.614, respectively, and the NDCG@10 values are 0.907 and 0.848. These results suggest that the rewritten job descriptions maintain, and in some cases slightly improve, the quality of the recommendations.

However, it is worth noting that for the profession of software engineer, there is a slight decrease in the match qualities for the generated descriptions compared to the original ones. This can be attributed to the highly skewed distribution of the dataset for this profession, with an 84:16 ratio favoring males. Despite this, the overall impact on the recommendation quality remains minimal.

These findings demonstrate that our approach to promoting fairness in job advertisements does not compromise the utility of matching with the best and most qualified candidates. The rewritten job descriptions maintain comparable recommendation quality while addressing potential biases and promoting diversity in the candidate pool.

Profession	Type	MRR@10	NDCG@10	MRR@25	NDCG@25	MRR@50	NDCG@50
Accountant	Original	0.614	0.848	0.614	1.466	0.620	2.265
	Generated	0.614	↑.059 0.907	↑.007 0.621	↑.052 1.518	↑.001 0.621	↓.020 2.245
Attorney	Original	0.709	1.212	0.709	2.148	0.709	3.334
	Generated	↑.028 0.737	↑.057 1.269	↑.028 0.737	↑.037 2.185	↑.028 0.737	↓.154 3.180
Journalist	Original	0.920	1.314	0.920	2.222	0.920	3.369
	Generated	↓.153 0.767	↓.034 1.280	↓.153 0.767	↓.097 2.125	↓.153 0.767	↓.252 3.117
Professor	Original	0.673	0.975	0.673	1.999	0.673	3.362
	Generated	↑.030 0.703	↓.054 0.921	↑.030 0.703	↓.134 1.865	↑.030 0.703	↓.135 3.227
Software Eng.	Original	0.086	0.140	0.092	0.275	0.097	0.472
	Generated	↓.020 0.066	↓.039 0.101	↓.017 0.075	↓.070 0.205	↓.018 0.079	↓.112 0.360

Table 4: Impact of fairness on recommendation quality across different professions and top-k values. Changes shown as absolute differences between original and generated versions.

Discussion

Our work demonstrates a practical approach to implementing AI governance principles in the context of automated hiring systems, showcasing how technical solutions can support fairness and inclusivity while maintaining system effectiveness. This research bridges a critical gap between AI policy goals and technical implementation, particularly in the domain of algorithmic hiring where fairness considerations are of greatest importance.

The implications of our work extend in several important directions. First, it provides a concrete example of how large language models can be governed and aligned with societal values through automated feedback mechanisms, reducing reliance on human oversight while still maintaining accountability. Second, it demonstrates how technical solutions can help enforce policy objectives - in this case, fair hiring practices as outlined in regulations like New York Local Law 144. This alignment between technical implementation and policy requirements is crucial for effective AI governance.

Several limitations and considerations are important to note. While our methodology reduces the need for human intervention, it remains computationally expensive, highlighting the need to balance governance objectives with practical constraints. The choice of pre-trained model can significantly impact results, emphasizing the importance of model selection in governance frameworks. Additionally, our reliance on surrogate metrics rather than real-world outcomes points to the broader challenge of establishing appropriate evaluation criteria for AI governance mechanisms. The potential for factual errors in the generated descriptions also underscores the ongoing need for human oversight in AI systems.

Acknowledgments

This work was funded by European Union's Horizon Europe research and innovation programme under grant agreement no. 101070568 (AutoFair).

References

Askell, A.; Bai, Y.; Chen, A.; Drain, D.; Ganguli, D.; Henighan, T.; Jones, A.; Joseph, N.; Mann, B.; DasSarma, N.; Elhage, N.; Hatfield-Dodds, Z.; Hernandez, D.; Kernion, J.; Ndousse, K.; Olsson, C.; Amodei, D.; Brown, T.; Clark, J.; McCandlish, S.; Olah, C.; and Kaplan, J. 2021. A General Language Assistant as a Laboratory for Alignment. arXiv:2112.00861.

Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; Chen, C.; Olsson, C.; Olah, C.; Hernandez, D.; Drain, D.; Ganguli, D.; Li, D.; Tran-Johnson, E.; Perez, E.; Kerr, J.; Mueller, J.; Ladish, J.; Landau, J.; Ndousse, K.; Lukosuite, K.; Lovitt, L.; Sellitto, M.; Elhage, N.; Schiefer, N.; Mercado, N.; DasSarma, N.; Lasenby, R.; Larson, R.; Ringer, S.; Johnston, S.; Kravec, S.; Showk, S. E.; Fort, S.; Lanham, T.; Telleen-Lawton, T.; Conerly, T.; Henighan, T.; Hume, T.; Bowman, S. R.; Hatfield-Dodds, Z.; Mann, B.; Amodei, D.; Joseph, N.; McCandlish, S.; Brown, T.; and Kaplan, J. 2022. Constitutional AI: Harmlessness from AI Feedback. arXiv:2212.08073.

Barikeri, S.; Lauscher, A.; Vulić, I.; and Glavaš, G. 2021. RedditBias: A Real-World Resource for Bias Evaluation and Debiasing of Conversational Language 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)*, 1941–1955.

Bender, E. M.; Gebru, T.; McMillan-Major, A.; and Shmitchell, S. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, 610–623.

Bolukbasi, T.; Chang, K.-W.; Zou, J. Y.; Saligrama, V.; and Kalai, A. T. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29.

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.

Casper, S.; Davies, X.; Shi, C.; Gilbert, T. K.; Scheurer, J.; Rando, J.; Freedman, R.; Korbak, T.; Lindner, D.; Freire, P.; Wang, T.; Marks, S.; Segerie, C.-R.; Carroll, M.; Peng, A.; Christoffersen, P.; Damani, M.; Slocum, S.; Anwar, U.; Siththaranjan, A.; Nadeau, M.; Michaud, E. J.; Pfau, J.; Krasheninnikov, D.; Chen, X.; Langosco, L.; Hase, P.; Bıyık, E.; Dragan, A.; Krueger, D.; Sadigh, D.; and Hadfield-Menell, D. 2023. Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback. arXiv:2307.15217.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Davani, A.; Díaz, M.; Baker, D.; and Prabhakaran, V. 2024. Disentangling Perceptions of Offensiveness: Cultural and Moral Correlates. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24, 2007–2021. New York, NY, USA: Association for Computing Machinery. ISBN 9798400704505.

De-Arteaga, M.; Romanov, A.; Wallach, H.; Chayes, J.; Borgs, C.; Chouldechova, A.; Geyik, S.; Kenthapadi, K.; and Kalai, A. T. 2019. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *proceedings of the Conference on Fairness, Accountability, and Transparency*, 120–128.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Elazar, Y.; and Goldberg, Y. 2018. Adversarial Removal of Demographic Attributes from Text Data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 11–21. Brussels, Belgium: Association for Computational Linguistics.

Fabris, A.; Baranowska, N.; Dennis, M. J.; Hacker, P.; Saldivar, J.; Borgesius, F. Z.; and Biega, A. J. 2023. Fairness and Bias in Algorithmic Hiring. *arXiv preprint arXiv:2309.13933*.

Garimella, A.; Amarnath, A.; Kumar, K.; Yalla, A. P.; Anandhavelu, N.; Chhaya, N.; and Srinivasan, B. V. 2021. He is very intelligent, she is very beautiful? on mitigating social biases in language modelling and generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 4534–4545.

Gira, M.; Zhang, R.; and Lee, K. 2022. Debiasing pre-trained language models via efficient fine-tuning. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, 59–69.

Guo, Y.; Yang, Y.; and Abbasi, A. 2022. Auto-debias: Debiasing masked language models with automated biased prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1012–1023.

Islam, S. M.; Nagpal, A.; Ganesan, B.; and Lohia, P. K. 2021. Fair Data Generation using Language Models with Hard Constraints. In *Annual Conference on Neural Information Processing Systems*.

Jiang, A. Q.; Sablayrolles, A.; Roux, A.; Mensch, A.; Savary, B.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Hanna, E. B.; Bressand, F.; et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Kaneko, M.; and Bollegala, D. 2021. Debiasing Pre-trained Contextualised Embeddings. In Merlo, P.; Tiedemann, J.; and Tsarfaty, R., eds., *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 1256–1266. Online: Association for Computational Linguistics.

Korbak, T.; Shi, K.; Chen, A.; Bhalerao, R.; Buckley, C. L.; Phang, J.; Bowman, S. R.; and Perez, E. 2023. Pretraining language models with human preferences. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Liang, P. P.; Li, I. M.; Zheng, E.; Lim, Y. C.; Salakhutdinov, R.; and Morency, L.-P. 2020. Towards Debiasing Sentence Representations. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5502–5515. Online: Association for Computational Linguistics.

Liu, X.; Shin, H.; and Burns, A. C. 2021. Examining the impact of luxury brand's social media marketing on customer engagement: Using big data analytics and natural language processing. *Journal of Business research*, 125: 815–826.

Liu, Y.; Han, T.; Ma, S.; Zhang, J.; Yang, Y.; Tian, J.; He, H.; Li, A.; He, M.; Liu, Z.; Wu, Z.; Zhu, D.; Li, X.; Qiang, N.; Shen, D.; Liu, T.; and Ge, B. 2023. Summary of ChatGPT/GPT-4 Research and Perspective Towards the Future of Large Language Models. arXiv:2304.01852.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692.

Lyu, C.; Xu, J.; and Wang, L. 2023. New trends in machine translation using large language models: Case examples with ChatGPT. *arXiv preprint arXiv:2305.01181*.

Mao, Y.; Yu, L.; Yang, Y.; Zhou, F.; and Zhong, T. 2023. Debiasing Intrinsic Bias and Application Bias Jointly via Invariant Risk Minimization (Student Abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 16280–16281.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781.

Morabito, R.; Kabbara, J.; and Emami, A. 2023. Debiasing should be Good and Bad: Measuring the Consistency of Debiasing Techniques in Language Models. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Findings of the Association for Computational Linguistics: ACL 2023*, 4581–

4597. Toronto, Canada: Association for Computational Linguistics.

Nadeem, M.; Bethke, A.; and Reddy, S. 2021. StereoSet: Measuring stereotypical bias in pretrained language 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)*, 5356–5371. Online: Association for Computational Linguistics.

Nangia, N.; Vania, C.; Bhalerao, R.; and Bowman, S. R. 2020. CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1953–1967. Online: Association for Computational Linguistics.

New York City Department of Consumer and Worker Protection. 2023. Automated Employment Decision Tools. https://rules.cityofnewyork.us/rule/automated-employment-decision-tools-updated/.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P. F.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 27730–27744. Curran Associates, Inc.

Pennington, J.; Socher, R.; and Manning, C. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. Doha, Qatar: Association for Computational Linguistics.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237. New Orleans, Louisiana: Association for Computational Linguistics.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Ravfogel, S.; Elazar, Y.; Gonen, H.; Twiton, M.; and Goldberg, Y. 2020. Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7237–7256. Online: Association for Computational Linguistics.

Sakaguchi, K.; Bras, R. L.; Bhagavatula, C.; and Choi, Y. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9): 99–106.

Sallam, M. 2023. ChatGPT utility in healthcare education, research, and practice: systematic review on the promising perspectives and valid concerns. In *Healthcare*, volume 11, 887. MDPI.

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

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347.

Si, C.; Friedman, D.; Joshi, N.; Feng, S.; Chen, D.; and He, H. 2023. Measuring Inductive Biases of In-Context Learning with Underspecified Demonstrations. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 11289–11310. Toronto, Canada: Association for Computational Linguistics.

Snell, C. V.; Kostrikov, I.; Su, Y.; Yang, S.; and Levine, S. 2023. Offline RL for Natural Language Generation with Implicit Language Q Learning. In *The Eleventh International Conference on Learning Representations*.

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

Taylor, R.; Kardas, M.; Cucurull, G.; Scialom, T.; Hartshorn, A.; Saravia, E.; Poulton, A.; Kerkez, V.; and Stojnic, R. 2022. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*.

Team, G. 2023. Gemini: A Family of Highly Capable Multimodal Models. arXiv:2312.11805.

Thoppilan, R.; De Freitas, D.; Hall, J.; Shazeer, N.; Kulshreshtha, A.; Cheng, H.-T.; Jin, A.; Bos, T.; Baker, L.; Du, Y.; et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Woods, A.; Tharakan, S.; and Brown, J. 2021. *Writing Inclusive Job Descriptions and Candidate Communication*, chapter 5. Wiley.

Yuan, H.; Yuan, Z.; Tan, C.; Wang, W.; Huang, S.; and Huang, F. 2023. RRHF: Rank Responses to Align Language Models with Human Feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zhang, Y.; Sun, S.; Galley, M.; Chen, Y.-C.; Brockett, C.; Gao, X.; Gao, J.; Liu, J.; and Dolan, B. 2020. DIALOGPT : Large-Scale Generative Pre-training for Conversational Response Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics:*

System Demonstrations, 270–278. Online: Association for Computational Linguistics.

Zhong, M.; Liu, Y.; Yin, D.; Mao, Y.; Jiao, Y.; Liu, P.; Zhu, C.; Ji, H.; and Han, J. 2022. Towards a Unified Multi-Dimensional Evaluator for Text Generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2023–2038.

Zmigrod, R.; Mielke, S. J.; Wallach, H.; and Cotterell, R. 2019. Counterfactual Data Augmentation for Mitigating Gender Stereotypes in Languages with Rich Morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1651–1661.

Related Work

Our work relates to several lines of previous research.

Fair representations. Early efforts in bias mitigation targeted removing gender biases in static embeddings where the semantic representation of a word is confined to just one vector like GloVe (Pennington, Socher, and Manning 2014) and Word2Vec (Mikolov et al. 2013) to achieve unbiased representations and word associations. Bolukbasi et al. (2016) studied how gender identity words are associated with specific occupations and subtracted the gender direction from word embeddings to neutralize the language while sustaining the equal distance between gender-neutral words and gendered pairs of words. Ravfogel et al. (2020) proposed INLP for debiasing embeddings through iterative null-space projections for guarding protected attributes. As the field evolved, the focus shifted towards debiasing contextual embeddings, such as ELMo (Peters et al. 2018). Kaneko and Bollegala (2021); Liang et al. (2020) highlighted the relative complexity of contextual embeddings compared to their static counterparts and the challenge of identifying which parameters contribute to the bias. Liang et al. (2020) proposed SENT-DEBIAS applicable at sentence levels to debias pre-trained contextual embeddings, where Kaneko and Bollegala (2021) developed both token and sentence level approach and emphasized the trade-off between accuracy and unbiasedness in such models.

Several methods, such as adversarial learning (Elazar and Goldberg 2018; Sakaguchi et al. 2021) and counterfactual data augmentation (Zmigrod et al. 2019; Islam et al. 2021), have been proposed to mitigate language models propagating biases present in a training corpus. Elazar and Goldberg (2018) investigated the encoding of demographic information in the intermediate representations learned by text-based neural classifiers. Zmigrod et al. (2019) aim to reduce bias by data augmentation through counterfactual statements. However, using augmented datasets is expensive and is prone to introduce noise and unrealistic scenarios that can negatively impact performance.

Debiasing LLMs. Gira, Zhang, and Lee (2022) proposed an LLM fine-tuning method by leveraging GPT-2 as an example, and showed their approach reduced the gender bias in GPT-2 on the StereoSet benchmark. While their method is relatively cost-effective compared to pre-training with an augmented dataset, the side effect of the fine-tuning for bias on downstream applications of language models has not been studied for their approach. Schick, Udupa, and Schütze (2021) developed a self-debiasing (SD) approach where a fine-tuned GPT-2 model self-diagnoses the bias and remove from the generated text. Garimella et al. (2021) suggested a combined method of further pre-training with fairness-aware datasets and then fine-tuning based on a loss function including regularizers for bias for BERT (Devlin et al. 2019). Mao et al. (2023) noted the gap in existing works which separates the fine-tuning for debiasing then fine-tuning for downstream applications. Authors named the former bias as *intrinsic bias* whereas the latter has been called as *application bias* which our work targets to solve.

Measuring bias in LLMs. A variety of benchmarks have been published to fairly evaluate and compare developed debiasing techniques aiming to address biases related to stereotypes or gendered word associations (Nadeem, Bethke, and Reddy 2021; Nangia et al. 2020). Barikeri et al. (2021) targeted the bias in LLMs fine-tuned to conversational dialogue and have introduced RedditBias, a dataset rooted in actual Reddit conversations, designed to measure and mitigate biases in conversational models across gender, race, religion, and queerness. The study benchmarks the DialoGPT (Zhang et al. 2020) model with this dataset, revealing biases, particularly towards religious groups, and demonstrates that certain debiasing techniques can address these biases without sacrificing model performance.

Prompt-based mitigation. Recent studies investigated prompt-based fine-tuning strategies. Guo, Yang, and Abbasi (2022) proposed extracting prompts from the pre-trained LLM (Devlin et al. 2019; Liu et al. 2019), through a beam-search style algorithm and then applying an equalizing loss over predicted token distributions. Their work demonstrated that the proposed bias mitigation strategy does not adversely impact the performance of LLM on downstream applications. Morabito, Kabbara, and Emami (2023) proposed an instructive-debiasing (ID) algorithm where debiasing prompts include specific information on the category of bias represented in the given text. Si et al. (2023) studied the effect of inductive biases through demonstrations without the model update in LLMs to overcome prior biases in an LLM. They conclude that intervention via inductive biases could be a helpful strategy to reduce the influence of feature biases, however, overcoming strong prior biases remains a challenging question on the topic of in-context learning.

Fine-tuning. Extensive research exists on the topic of model fine-tuning (Askell et al. 2021; Yuan et al. 2023). Ouyang et al. (2022) and Liu et al. (2023) underscore the importance of RLHF in the fine-tuning process. While the models fine-tuned with RLHF have shown promise in generating more truthful and less toxic outputs, challenges persist. Ouyang et al. (2022) emphasizes the balance between model alignment with human intent and maintaining high performance, highlighting the complexities and nuances of leveraging RLHF in the fine-tuning process. Yuan et al. (2023) offers a critique on the RLHF approach, particularly highlighting the complexities associated with the PPO method (Schulman et al. 2017). Casper et al. (2023) survey the challenges and limitations of RLHF, specifically noting the difficulties attached to human evaluators, data quality and limitations of feedback types. Our approach addresses the concerns related to human evaluators as such eliminating the need for human feedback in fine-tuning. In some cases, (Bai et al. 2022; Rafailov et al. 2023) LLMs are themselves used to produce alignment data used in fine-tuning. However, this can inadvertently reintroduce fairness and bias issues into the aligned LLMs sourcing from AI feedback.

From the hiring domain perspective, there is a large literature on fairness challenges in algorithmic hiring. We point to a recent survey (Fabris et al. 2023).

Experiment Details

Datasets

Source	No. Jobs	No. profiles
Hacker News	76,000	20,300
Bias in Bios	76,000	50,000
Hiring Platform	5,745	1,000,000

Table 5: Summary of datasets used in experiments

Hacker News This dataset encapsulates a diverse collection of hiring posts from the tech news platform, Hacker News, with various splits, including “hiring” (76K posts) and “wants_to_be_hired” (20.3K profiles). Each entry in the dataset contains a “text” field, representing the content of the post, which corresponds to the job post in the “hiring” split and candidate profiles in the “wants_to_be_hired” split. We curated a final version of the dataset for training, extracting specific useful features from the original job descriptions. The details of data processing and the implementation of ILQL, including hyperparameters, are detailed below. We also conducted an analysis of the differential impact of gender identification on our embedding-based recommendation engine with Hacker News data. The results demonstrated that some roles are more susceptible to gender biases, details can be seen in Figure 2.

Bias in Bios (Candidates) We created a new dataset using a subset of the online biographies dataset introduced by De-Arteaga et al. (2019). This dataset consists of over 400K biographies collected from the Common Crawl corpus, labelled with 28 different occupations. For our experiments, we focused on a subset of the test split, selecting biographies corresponding to occupations commonly found in job advertisements on Hacker News. This resulted in a candidate pool of 50K profiles. We then matched this candidate pool with both original and generated job descriptions from Hacker News and computed various diversity metrics, including the True Positive Rate Gender Gap (TPR-GAP). In this context, we consider true positive predictions as correct occupation matches between the labelled occupation of candidates and the occupations mentioned in the job advertisements, computed separately for each gender.

Hiring Platform Data. The third dataset is from a live hiring platform. This dataset consists of 5745 job openings from around the world, with a very large pool of candidate profiles, obtained from publicly available data sources. Each job advertisement contains the job title, the description, and the requirements of the job. The jobs are distributed across different functions (e.g. engineering, accounting) and industries (e.g. software companies, food service). Each job is accompanied by a pool of 3K candidate profiles, both relevant and irrelevant to the job. The similarity of a candidate profile to the job is a float in $[0, 1]$ (higher values correspond to better matching candidates). For our experiments, we sample one million candidate profiles from this dataset randomly and use this subset as our common recommendation pool.

Processing Details

Hacker News Dataset We processed the original Hacker News dataset[5] to make it more informative, structured and compatible with our environment. The details of processing for both job descriptions and candidate profiles are given next.

Job descriptions: The 'text' column of the original 'hiring' split includes job descriptions in an unstructured text format. We extract several features from this text to generate prompts to guide our approach to rewriting job descriptions while keeping the important job specifications in the rewritten text. We consider the job title, location of the opening, required technologies, the company offering the position, and if the remote working option is available as important features to use in the prompt. The template of 'prompt' is as follows:

> *Original job description for reference:* `<text>`*. Based on this, the job is in* `<location>`*, at* `<company>` *for the* `<job title>` *position. The ideal candidate is skilled in* `<technologies>`*.* `<Remote statement>` *Write a new job description using only the original information.*

Then, we calculated the diversity score of each job description using the recommender system, illustrated in Figure 2.

Candidate profiles: The 'text' column of the original 'wants_to_be_hired' split includes candidate profiles in a relatively better-structured text format compared to job postings. While the dataset includes the location information of candidates, it doesn't contain any information related to gender which is an important feature to analyze the bias. Hence, we randomly assigned a gender for each candidate profile. The distribution of candidates based on geolocation and assigned genders is given in Table 6.

The embeddings of the both job descriptions and candidate profiles have been extracted from BERT (Devlin et al. 2019) to use in recommendation engine. The splits with their engineered features are given in Table 7.

Hiring Platform dataset Data from the hiring platform consists of job profiles (Table 8) and candidate profiles (Table 9). The dataset consists of 5,745 job descriptions and several million candidate profiles from which we sample one million profiles for our experiments. The dataset is fully annoymised including the masking of institution names for experience and education history.

The gender of candidates is reported as male (43.6%), female (34.2%), or unknown (22.2%). This forms the target gender distribution as this is considered to be the applicant pool. Similarly, the location of candidates is available at the country level. This is aggregated by continent. The observed frequencies of candidates in this dataset serve as target location distribution. The reward model is additive in the Wasserstein distances for gender and location.

The downstream task is one of matching candidates to job openings. For this we order candidates based on cosine similarity between the embedding of job description and candidate description. The candidate description is generated based on education and experience data on the candidate using templates. Both job and candidate embeddings are generated using a pre-trained model BERT.

The job descriptions are partitioned into train and 10% reserved for testing. The train data is further partitioned with 10% kept for validation. For our experiments, all one million candidates are considered to be viable for all job postings. In practice, this is not the case, as there may be filtering rules in place that limit the scope of the targeted audience. These constraints were not imposed.

For evaluation, we use the measures described in the main paper. Impact ratios, as codified in New York local laws, are retrospective, i.e. aim to audit hiring practices by examining hiring of candidates for each cohort. In our example, the hiring decision has not yet taken place. We consider a "success" if a candidate is ranked in the top ten for each job position. The denominator of the impact ratio, i.e. the applicant pool, is considered to be the top-50 applicants.

Impact Ratios for Geolocation

Implementation

We implemented the ILQL approach as detailed in (Snell et al. 2023), using GPT-2 as our pre-trained LLM. Initially, GPT-2 was fine-tuned with original job descriptions to emulate the relationship between the provided prompt and generate detailed job descriptions. Subsequently, an RL agent was trained to determine the Q-value of the generated text and adjust subsequent token probabilities to meet diversity objectives. The 'hiring dataset' was split into training, test, and evaluation subsets, while the 'wants_to_be_hired' dataset was used to compute diversity scores during both training and evaluation. To reduce the training time, 10% of the hiring dataset has been used in experiments instead of entire dataset.

For training, job descriptions were limited to 256 tokens, and the generated text was similarly restricted to 256 tokens. Within the recommender system, we set k to 50. During inference, we selected $\beta = 64$, and results for varying β values can be found in the next section. Our fine-tuned model was trained for 7 epochs with a learning rate of 1e-3, and the perturbation model was trained for an additional 7 epochs with the same learning rate. Altogether, the process took 10 hours using 8 V100 GPUs.

Hyperparameter search

We have reported the results and examples for $\beta = 8$ in Results section. This value has been chosen intuitively with few trials in experiments. Here, we share the results for varying β values in Table 11. The results in Table 11 demonstrates that changes in β do not cause significant deviations in the evaluations and our approach consistently outperforms the original dataset in terms of diversity score in all evaluated values of β hyperparameter.

[5] https://huggingface.co/datasets/dansbecker/hackernews_hiring_posts

[6] https://huggingface.co/deepset/roberta-base-squad2

	Gender		Geolocation							
	Female	Male	NA	Europe	SA	Asia	Africa	Remote	Australia	Unknown
Original	0.5	0.5	0.55	0.21	0.03	0.1	0.01	0.01	0.01	0.06

Table 6: Distribution of candidate profiles over genders and geolocations. NA and SA represents north and south America, respectively.

Dataset	Feature	Extraction Method
hiring	Job title	QA: What is the job title in the text?
hiring	Location	QA: What are the locations in the text?
hiring	Technologies	QA: What are the technologies in the text?
hiring	Company	QA: What is the company name in the text?
hiring	Remote	Text processing
hiring	Embedding	Extracted from BERT with given candidate profile
hiring	Prompt	Template: Job details (see caption)
hiring	Q-value	Diversity score obtained through recommendation engine
wants_to_be_hired	Gender	Randomly assigned
wants_to_be_hired	Location	Text processing
wants_to_be_hired	Embedding	Extracted from BERT with given candidate profile
wants_to_be_hired	Remote	Text processing

Table 7: Extracted features to obtain the final version of the dataset. QA represents the feature extraction methods using question-answering with Roberta[6], where the context is always the original job description.

Variable Name	**Description**	**Data Type**	**Example Values**
id	Unique identifier for the job	Integer	0
account_id	Unique identifier of the account (company) that posted the job ad	Integer	23
title	The title of the job	str	Front-end Developer
required_experience	The level of experience required for the particular job	str	Mid-Senior level
required_education	The education level required for the particular job	str	Bachelor's Degree
remote	Whether the job is remote or not	bool	FALSE
employment_type	The employment type	str	Full-time
industry	The industry of the company that published the job ad	str	Staffing and Recruiting
function	The function of the particular job	str	Engineering
detailed_location	The location of the company	JSON	{”country_code”: [str] ”IT”, ”state_code”: [str] ”MI”, ”city”: [str] ”Milan”, ”sub-region”: [str] “Metropolitan City of Milan”, ”zip_code”: [str] “11111”}
description	The description of the job, containing details about the role	str	”We are looking for a ...”
requirement_summary	The requirements of the job	str	”Proven experience as ...”
benefit_summary	The benefits that the company will provide to the hired candidate(s)	str	”- Health Care Plan ...”

Table 8: Hiring Platform Dataset - Job Advertisement.

Variable Name	**Description**	**Data Type**	**Example Values**
id	Unique identifier for the profile	Integer	0
job_id	The ID of the job that the candidate corresponds to in the dataset	Integer	0
country_code	The country code of the residence of the candidate	str	UK
gender	The predicted gender of the candidate	str	Male, Female, Unknown
experiences	The work experiences of the candidate	JSON	{”company”: [int] 99, ”start_date”: [str] ”2020”, ”end_date”: [str] “2022”, ”title”: [str] ”Graphic Designer”}
educations	The educations of the candidate	JSON	{”school”: [int] 10, ”start_date”: [str] “2016”, ”end_date”: [str] ”2020”, ”field_of_study”: [str] ”Design”, ”degree”: [str] ”Bsc”},
industry	The industry where the candidate has worked at	str	Computer Software
function	The function of the latest work experiences of the candidate	str	Engineering

Table 9: Hiring Platform Dataset - Profiles.

	Geolocation					
	IR_{NA}	IR_{Eu}	$\text{IR}_{\text{Africa}}$	IR_{Asia}	IR_{SA}	$\text{IR}_{\text{Remote}}$
Original	0.84±0.32	0.16±0.33	0.0±0.0	0.11±0.30	0.05±0.21	0.01±0.11
GPT-2-large	0.84±0.31	0.15±0.33	0.0±0.0	0.1±0.3	0.07±0.24	0.02±0.11
+INLP-race	0.87±0.29	0.17±0.35	0.01±0.08	0.08±0.26	0.06±0.24	0.01±0.07
+INLP-gender	0.87±0.29	0.17±0.35	0.01±0.08	0.08±0.26	0.06±0.24	0.01±0.07
+SentD-race	0.87±0.29	0.17±0.35	0.01±0.08	0.08±0.26	0.06±0.24	0.01±0.07
+SentD-gender	0.86±0.3	0.17±0.35	0.01±0.08	0.08±0.26	0.06±0.24	0.01±0.1
+SD	0.83±0.32	0.18±0.36	0.0±0.05	0.11±0.3	0.05±0.22	0.01±0.11
+ID	0.84±0.32	0.14±0.33	0.0±0.02	0.13±0.32	0.05±0.21	0.01±0.11
+AutoRefine	0.83±0.31	0.17±0.34	0.1±0.08	0.12±0.32	0.02±0.12	0.03±0.16

Table 10: Geolocation-specific comparison of generated job description results. The reported score is the mean and standard deviation of the metrics over the evaluation dataset.

	Diversity score	Gender	
		$\text{IR}_{\text{female}}$	IR_{male}
Original	-23.48±16.31	0.84±0.27	0.76±0.34
Rewrite ($\beta = 2$)	-22.25±16.95	0.83±0.28	0.78±0.32
Rewrite ($\beta = 4$)	-22.17±17.00	0.83±0.23	0.78±0.32
Rewrite ($\beta = 8$)	-20.17±14.49	0.84±0.25	0.78±0.30
Rewrite ($\beta = 16$)	-22.05±17.02	0.84±0.26	0.78±0.32
Rewrite ($\beta = 32$)	-22.02±17.02	0.84±0.27	0.78±0.32
Rewrite ($\beta = 64$)	-21.10±17.20	0.85±0.25	0.76±0.33
Rewrite ($\beta = 128$)	-22.19±16.99	0.83±0.27	0.78±0.32

	Geolocation					
	IR_{NA}	IR_{Eu}	$\text{IR}_{\text{Africa}}$	IR_{Asia}	IR_{SA}	$\text{IR}_{\text{Remote}}$
Original	0.84±0.32	0.16±0.33	0.0±0.0	0.11±0.30	0.05±0.21	0.01±0.11
Rewrite ($\beta = 2$)	0.82±0.33	0.17±0.35	0.0±0.0	0.10±0.30	0.03±0.17	0.04±0.20
Rewrite ($\beta = 4$)	0.79±0.35	0.17±0.36	0.0±0.0	0.12±0.32	0.04±0.18	0.04±0.19
Rewrite ($\beta = 8$)	0.83±0.31	0.17±0.34	0.1±0.08	0.12±0.32	0.02±0.12	0.03±0.16
Rewrite ($\beta = 16$)	0.81±0.34	0.16±0.34	0.0±0.0	0.11±0.31	0.03±0.18	0.04±0.19
Rewrite ($\beta = 32$)	0.80±0.35	0.18±0.36	0.0±0.0	0.11±0.31	0.03±0.15	0.04±0.19
Rewrite ($\beta = 64$)	0.81±0.33	0.19±0.37	0.0±0.0	0.12±0.31	0.03±0.16	0.03±0.15
Rewrite ($\beta = 128$)	0.81±0.34	0.16±0.35	0.0±0.0	0.10±0.29	0.04±0.18	0.04±0.20

Table 11: Comparison of generated job description results with scores of original job descriptions for varying values of β.

Differential Impact of Gender Identification on Recommendation Engine

To ensure that our model successfully generates unbiased job descriptions, we have devised a systematic evaluation approach focused on quantitatively measuring how well the descriptions align with diversity goals. For the evaluation, we first examined original job descriptions to discern which roles were most susceptible to biases. For the candidate recommendation phase, two distinct profiles were constructed for every candidate:

1. The gendered profile: This contains a clear statement of gender, articulated as “I identify as {gender}.”
2. The gender-neutral profile: This profile is taken from the original dataset without any explicit gender markers.

Both of these profiles were utilized in the candidate-matching phase with original job descriptions. This approach enabled us to ascertain how gendered or gender-neutral embeddings influenced the recommendation system. We proceeded to determine the disparity in gender distribution (between female and male candidates) matched to each job title utilizing both candidate profile sets of embeddings. The job roles most impacted by gendered embeddings are shared in Figure 2.

Introducing Language Quality into Reward Function

In addition to the reward function presented in Equation 8, we investigated the efficacy of incorporating language quality into the reward function. We adopted the metrics suggested by (Zhong et al. 2022) to assess the coherence, fluency, and relevance of the generated text. Consequently, we define the reward function $\mathcal{R}$ for each generated job description x as:

$$\mathcal{R}(y_p^*) = \text{LQS}(y_p^*) - \lambda \left(\Delta_{\text{gender}}(y_p^*) - \Delta_{\text{geolocation}}(y_p^*) \right) \tag{11}$$

Here, LQS denotes the language quality metrics, while Δ_{gender} and $\Delta_{\text{geolocation}}$ represent the elements of the diversity score. The parameter λ balances language quality with inclusivity considerations. Initially, we fine-tuned the pre-trained GPT-2 for 7 epochs and subsequently trained the perturbation model (RL agent) for 70 epochs. The entire training process spanned 3 days on 8 V100 GPUs. Achieving model convergence with the language score was notably slower than with our primary experimental setup. This delay was attributed to the computational demands of integrating language quality evaluations into the reward function. We showcase the diversity score and impact ratio outcomes for $\beta = 64$ and $\lambda = 1$ in Table 12. Our findings indicate that, given the computational costs associated with the two reward functions and the lack of significant improvements in the impact ratio and diversity score, it is more advantageous to solely use the diversity score as a reward function.

Examples

We show some example rewrites along with associated metrics in Tables 13 and 14. In these cases, the rewritten text changes key attributes of the description of the opening (e.g. “cells and molecules” to “databases and documents”, “Field” to “System”), company (e.g. “NVIDIA” to “MIT”) or stakeholder names (“donor” to “investor”), or locations (e.g. “Princeton” to “Paris”) and negations (e.g. “wont” to “would”). This introduces factual errors in the text. For our test sample of 400 job descriptions, these were relatively infrequent. However, we have not systematically measured

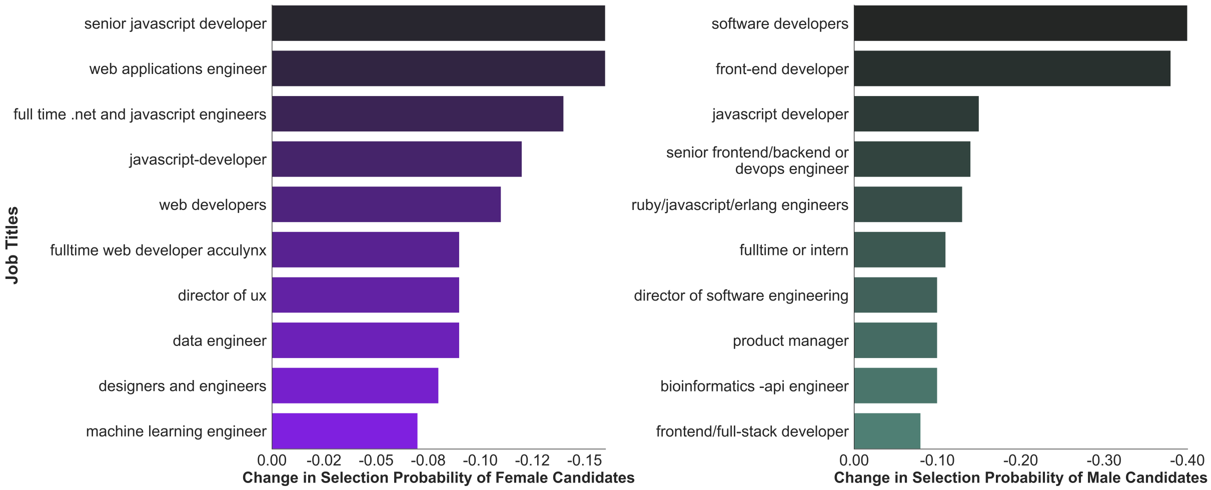

Figure 2: Differential impact on selection probabilities across job titles by gender. This figure visualizes the changes in selection probabilities for various job titles when gender identification is incorporated into candidate profiles. The depicted titles are those experiencing the most pronounced shifts in probabilities. Negative values indicate a reduction in selection probability.

	Diversity score	Gender	
		$\text{IR}_{\text{female}}$	IR_{male}
Original	-23.48±16.31	0.84±0.27	0.76±0.34
Rewrite	-21.97±16.92	0.83±0.27	0.81±0.31

	Geolocation					
	IR_{NA}	IR_{Eu}	$\text{IR}_{\text{Africa}}$	IR_{Asia}	IR_{SA}	$\text{IR}_{\text{Remote}}$
Original	0.84±0.32	0.16±0.33	0.0±0.0	0.11±0.30	0.05±0.21	0.01±0.11
Rewrite	0.10±0.29	0.07±0.25	0.02±0.12	0.10±0.29	0.07±0.25	0.02±0.12

Table 12: The results of training with reward function in Equation 1.

Description	Before			After		
	DS	IR_m	IR_f	DS	IR_m	IR_f
It'd be a big plus if you have: experience developing games; full health, dental, vision coverage; -~~snacked-filled kitchen and booster juice breaks;~~ catered breakfast, lunch, and dinner; - convenient location downtown Toronto	-25.35	0.44	1.00	-11.35	0.25	1.00
XXX, located ... As compensation, we're offering a competitive salary, ..., snacks on snacks ~~-on snacks~~, daily catered lunch, ...	-19.35	0.59	1.00	-13.35	0.72	1.00
We are hiring exceptional engineers ... are funded by the ~~CEO~~ executives of Yelp, Dropbox, Yammer, Box, Parse, and others, as well as Google Ventures, Salesforce and Y-Combinator. Full list at ~~www.~~[REDACTED URL]. Payroll is complex and there are tough engineering challenges to be ~~tack~~ handled... We strive for 100% test coverage, and every commit is code reviewed~~ed by another developer on the team~~...	-19.35	0.48	1.00	-13.35	1.00	0.40
XXX~~\o/~~ - Palo Alto, CA - Full Time ... - H1B OK (visa sort~~ed~~) XXX captures and indexes every word spoken on TV... and are continuing our m~~arch~~ ove onto GoogleTV and connected devices	-23.35	1.00	0.61	-17.35	0.85	1.00
... we're a technology company ~~maniacally focused on~~ dedicated to a great product. Companies (that you've definitely heard of) use Str~~e~~ ongak everyday to make their teams more effective. Future founders, this is a great way to get real experience on what its like starting a company - on our ~~dim~~ note...~~Obvious~~ Unfortunately:... Our benefits package is amazing ~~We are very well funded...~~	-21.35	0.16	1.00	-15.35	0.78	1.00

Table 13: Examples of rewrites from the test set showing modifications made by the RL agent along with associated diversity score and impact ratio before and after edits. Relatively minor edits lead to gender inclusivity.

hallucinations.

The risk of hallucinations can be mitigated to a large extent as in our intended use case, the rewrite is presented to a recruiter and benefits from human oversight.

Description	Before			After		
	DS	IR_m	IR_f	DS	IR_m	IR_f
Envision a massive, fully-automated research facility that moves around, mixes, and analyzes ~~cells and molecule~~ and databases and documents and papers and things on a scale equivalent to millions of technicians doing the work by hand. We’ll call it the world’s first ~~”~~biological server farm~~”~~—biology will become a programming discipline, and biologists won’t need their own labs anymore. W~~ant to help us build it?~~ e’re looking for extremely talented software engineers from a variety of backgrounds. We’re ~~a~~ w~~ell-funded, stealth startup based in Menlo Pa~~ ork~~, founded by scientists and engineers who want to solve biology in their lifetimes. We’re looking for extremely talented software engineers from a variety of backgrounds. We’re working main~~ ing mostly with C++ and Python in a Linux environment.	-14.85	1.00	0.37	-9.85	1.00	0.62
Santa Clara, CA, Full-time, Linux kernel - Virtualization engineer at ~~NV~~ MI~~DIA~~ T. We are looking for talented embedded system software engineers with a focus on virtualization to help us architect next generation hypervisor software for ~~NVIDIA platforms~~ the Linux kernel. This is a position in Santa Clara, CA. Some of the skills we look for: Technical expertise on the ARM architecture, embedded virtualization, ~~multicore~~ divisionot designs, Linux kernel, device drivers and embedded software in general. P~~ractical understanding and implementation of microkernel~~ practical understanding and implementation of microkots, hypervisor design, m~~ulticore, cache coherency, concurrency, systems level API design, virtual memory management. Also development of virtualization interfaces for the Linux kernel. Key~~ icrokiwo~~rds/Specialties: Virtualization, hypervisor design, microkernel~~s, ARM ~~A~~ architecture, Linux kernel, virtual memory management, Multicore...	-25.35	1.00	0.76	-13.35	0.62	1.00
GiveNext - C~~leveland~~ ity, OH or REMOTE. GiveNext is the easiest way for ~~don~~ investors to give to the causes they care about. We support giving to 1.4 million nonprofits. Looking for a full-time technical cofounder / CTO. You’ll be paid a salary plus have stock options.	-16.35	0.92	1.00	-4.85	1.00	1.00
Daily Harvest - jobs: Software Engineer + more - New York City, NY or P~~rinceton~~ aris, NJ — Full-time Onsite — Everyone around you – especially the non-techies in your life – will at least try, if not consistently enjoy the ~~frozen superfood eats~~ rocket superfood that your work at Daily Harvest will deliver! Our 50+ flavor combination~~s of smoothies, o~~ of smoothies, instant oops, chia Parfaits, and Harve~~rnight oats, chia parfaits, and har~~ stbowl are co- created by our team of chefs and nutritionists and come packed with organic products and no added added sweet or presews....	-20.35	1.00	0.70	-9.85	1.00	0.72
We ~~intend~~ plan to popularize the production of custom gadgets all over the world. This summer internship is more like an apprenticeship where you learn the ~~rope~~ ways while following an experienced engineer. ... Monthly stipend (R~~s~~.291~~6~~7. If you applying from outside India, keep in mind that the total stipend wo~~nt~~uld cover your traveling costs.	-21.35	0.37	1.00	-11.35	0.43	1.00
*Consulting Engineer (~~Field~~ System/implementation/post-sale Engineers) Location: New York, NY / Washington D.C. (Clearance is required) As a technical consultant, you’ll be~~ MongoDB’s ambassador to our clients and other MongoDB users. You’ll deliver advisory consulting to and lead comprehensive training sessions with MongoDB’s clients, helping them solve mission-critical challenges in areas as varied as schema design, performance optimization (both in a database and in an application), software architecture, production operations. A development/distributed systems background is required.~~	-18.85	1.00	0.29	-9.85	1.00	0.62

Table 14: Additional examples with associated metrics.