

BiasScope: Inference-Time Bias Detection and Mitigation for Fair Natural Language Processing (NLP) Using Retrieval-Augmented Generation and Prompt Engineering

Rakeshkumarreddy Ambati
rakeshreddy0507@gmail.com
Independent Researcher

Abstract

The appearance of social stereotypes in Large Language Models (LLMs) is a crucial concern in research on fairness in the context of Natural Language Processing (NLP). This study focuses on gender bias, occupational stereotypes, gender comparisons, and trait attributions. Numerous mitigation techniques depend on fine-tuning and post-processing, which are typically expensive and difficult to audit. BiasScope is a lightweight framework that uses prompt engineering and RAG to detect and mitigate model biases at inference time without changing weights. We generate responses through the Gemini 2.5 Flash API in our implementation. The system (1) provides a baseline reply, followed by (2) scoring by a lexicon- and pattern-based detector with transparent functioning. When bias is detected, (3) it retrieves fairness guidance, followed by (4) generating a mitigated response from an augmented prompt. The mean composite bias score on the 26-prompt evaluation set is reduced by 72.1% after mitigation. This composite score combines four detector signals: toxicity, counterfactual sensitivity, stereotype co-occurrence, and sentiment and sentiment is treated as one signal among four, not as proof of bias. No formal user study of response quality was conducted; qualitative inspection of the outputs is reported in Section 4. BiasScope is designed to support multiple LLM backends and is fully traceable from detection to mitigation. It provides an effective means of achieving fairness and accountability in production Natural Language Processing (NLP).

Keywords

• Bias Mitigation • Large Language Models (LLMs) • Natural Language Processing (NLP) • Prompt Engineering • Retrieval-Augmented Generation (RAG) • Responsible Artificial Intelligence

1. Introduction

Large Language Models have shown impressive capabilities in various natural language processing tasks such as creative writing, code generation, reasoning, and question answering. As a result of being trained on large amounts of data using internet text, the technology has benefited many people, including through chatbots and other applications. Nonetheless, the strength of these models is also a challenge. Since social biases are present in the training corpora, learning from diverse and extensive datasets can lead to reproducing these biases [1].

Concerns regarding gender and occupational bias in LLM outputs are particularly pervasive. Studies have shown that these models usually associate certain professions with specific genders. For instance, engineering and surgical roles are linked to masculine pronouns, while nursing and teaching positions are associated with feminine reference. While these correlations may capture the historical distribution of

individuals in occupations, using them in hiring, education, healthcare communication, or legal support can perpetuate stereotypes with real consequences [2]. Even the subtle biases exhibited in outputs from these models can perpetuate unfair assumptions [3].

Current methods that tackle bias in LLMs can be broadly classified into pre-training interventions and post-training modifications. Pre-training debiasing means cleaning the training data of biased content or adapting the training to obtain fair representation. Methods such as fairness fine-tuning, counterfactual data augmentation, or output filtering can be used. Although these methods have been more or less effective, they have several common drawbacks such as high computational costs, limited reproducibility, and difficulty of auditing, apart from cases when only API access to models is possible [4].

The unique part of BiasScope is its lightweight, modular design that decouples bias detection from mitigation while maintaining full traceability. The framework is designed to support multiple LLM backends, though only the Gemini 2.5 Flash API was evaluated in this study. The bias detection mechanism determines bias using a scoring method that is based on a lexicon and patterns over a host of bias dimensions. Furthermore, it is ensured that the detection method is reproducible and transparent. When there is bias, the bias alert retrieves the relevant fairness guidelines from a knowledge source and appends them as additional context to the generation prompt [5]. In this way, practitioners will know not just that bias was detected but also which patterns triggered detection and which guidance was used to moderate it. Recent advances in retrieval-augmented generation have made such knowledge-grounded conditioning increasingly practical for production systems [6].

BiasScope: Architecture, Implementation and Evaluation of a New Tool. The system substantially minimizes bias across a pool of diverse prompts without sacrificing the quality and coherence of the responses. The rest of this paper is organized as follows: the methodology is presented in Section 2, which describes the theory as well as the technique. The system architecture and implementation are discussed in Section 3. The test and results are in Section 4. Section 5 outlines implications, limitations, and broader context. In Section 6, we conclude and offer future directions.

2. Related Work

Addressing bias in language models has been a central concern in responsible AI research. Early approaches focused on data-level interventions such as counterfactual data augmentation and reweighting, but these often require retraining and are computationally prohibitive for large models. Post-training mitigation techniques, including fine-tuning with fairness objectives and output constraints, have been explored but suffer from limited generalizability and lack of transparency. Recent work has shifted towards inference-time strategies that leverage prompt engineering and retrieval-augmented generation to guide model outputs without altering weights [7]. These methods are particularly attractive for production systems where model updates are costly and auditability is paramount.

Recent work has explored inference-time mitigation through self-diagnosis and self-debiasing, as well as retrieval-augmented generation for knowledge grounding and control [4]. Surveys of bias in NLP emphasize the need for transparent and interpretable evaluation [1]. Established benchmarks such as StereoSet [8], BBQ [9], and BOLD [10] provide external validation resources. Our work builds on these foundations by proposing a lightweight, modular framework that combines transparent bias detection with RAG-based mitigation, addressing the gaps in existing inference-time fairness approaches.

The use of RAG for fairness is still nascent; prior work has primarily focused on factual accuracy and knowledge grounding [5]. However, the idea of retrieving ethical guidelines or fairness principles to condition generation is gaining traction. Comprehensive surveys have catalogued the evolution and

current landscape of retrieval-augmented generation [11], while others have examined best practices for retrieval and augmentation [12]. Several surveys have catalogued bias sources and mitigation strategies, highlighting the need for practical, interpretable solutions [2]. Our work builds on these foundations by proposing a lightweight, modular framework that combines transparent bias detection with RAG-based mitigation, addressing the gaps in existing inference-time fairness approaches.

Recent research has also explored multi-paradigm AI frameworks for real-time anomaly detection and adaptive risk scoring, which share methodological similarities with our bias detection approach in their emphasis on transparent, rule-based scoring mechanisms [3]. Cloud computing architectures for multi-tenant LLM agents have addressed deployment considerations relevant to production fairness systems [13]. Software physics approaches for safe agentic AI-driven code evolution provide complementary perspectives on ensuring safety in AI systems. Additionally, model-driven engineering for health data management demonstrates the importance of integrating fairness considerations into domain-specific applications [14].

3. Methodology

The BiasScope framework operates under the premise that one can mitigate bias using prompt engineering and without changing the model parameters. The basis of this technique stems from the observation that LLMs are sensitive to contextual cues within prompts. Moreover, they are influenced, to a large degree, by explicit instructions on fairness. The methodology follows a three-part pipeline of detection, retrieval, and mitigation, each using distinct techniques to ensure effective and illuminative bias mitigation.

The detection phase uses a multi-dimensional scoring system that assesses the generated text over various bias indicators. This approach is computationally lightweight and interpretable, as opposed to less interpretable learned bias classifiers [3], and achieves a reasonable level of detection performance. The scoring system consists of four key components: toxicity (T), counterfactual sensitivity (C), stereotype (S), and sentiment (P). Each component produces a normalized score in the range $[0, 1]$, where 0 indicates no detectable bias signal and 1 indicates the maximum detectable bias signal for that dimension. The composite bias score B is defined as the weighted sum of the four component scores:

$$B = w_T T + w_C C + w_S S + w_P P \quad (1)$$

3.1 Detector Component Specifications

Toxicity lexicon and patterns. The toxicity component uses a curated lexicon $\mathcal{L}_{\text{tox}}$ of approximately 120 terms and phrases associated with toxic, derogatory, or demeaning language. The lexicon was constructed by combining entries from established toxicity resources and manually filtering for terms relevant to gender and occupational bias. In addition to direct term matching, the detector identifies contextual patterns using regular expressions that capture co-occurrence relationships, such as a toxic term appearing within a five-token window of a demographic group mention. The toxicity score T is computed as:

$$T = \min \left(1, \frac{n_{\text{tox}} + \alpha \cdot n_{\text{ctx}}}{N_{\text{tokens}}} \cdot \kappa \right) \quad (2)$$

where n_{tox} is the number of direct lexicon matches, n_{ctx} is the number of contextual pattern matches, $\alpha = 0.5$ weights contextual matches, N_{tokens} is the total token count of the response, and $\kappa = 100$ is a scaling constant. This approach to contextual sensitivity mitigates false positives that otherwise arise with naive lexicon-based methods [15].

Counterfactual procedure. Given a generated response R , the system produces a counterfactual response R' by swapping gendered terms and pronouns according to a predefined mapping $\mathcal{M}_{\text{gender}}$ (e.g., he $\leftrightarrow$ she, him $\leftrightarrow$ her, his $\leftrightarrow$ hers, man $\leftrightarrow$ woman, men $\leftrightarrow$ women, male $\leftrightarrow$ female, masculine $\leftrightarrow$ feminine). The output is not regenerated; the counterfactual is produced by deterministic lexical substitution on the original response. The sentiment difference is computed as:

$$C = \frac{|\text{sent}(R) - \text{sent}(R')|}{2} \quad (3)$$

where $\text{sent}(\cdot) \in [-1, 1]$ is the sentiment polarity score computed using a lexicon-based sentiment analyzer. The normalization by 2 maps the maximum possible difference (from -1 to $+1$) to the range $[0, 1]$. A threshold of $C > 0.1$ is used to flag counterfactual sensitivity as a contributing bias signal.

Stereotype mappings. The stereotype component uses a mapping $\mathcal{M}_{\text{stereo}}$ that associates demographic groups with stereotypical roles, traits, and occupations. The mapping is organized into three categories: (i) occupation-role associations (e.g., surgeon $\rightarrow$ male, nurse $\rightarrow$ female, engineer $\rightarrow$ male, teacher $\rightarrow$ female, CEO $\rightarrow$ male, secretary $\rightarrow$ female); (ii) trait attributions (e.g., male $\rightarrow$ assertive, decisive, logical; female $\rightarrow$ caring, empathetic, gentle); and (iii) comparison framings (e.g., “men are better at X”, “women are more Y”). Each mapping entry is represented as a triple (demographic group, stereotypical attribute, category). The stereotype score S is computed as:

$$S = \min \left(1, \frac{n_{\text{stereo}}}{N_{\text{mentions}}} \cdot \lambda \right) \quad (4)$$

where n_{stereo} is the number of co-occurrences between a demographic group mention and a mapped stereotypical attribute within a ten-token window, N_{mentions} is the total number of demographic group mentions, and $\lambda = 2$ is a scaling constant.

Sentiment calculation. The sentiment component computes the emotional polarity of demographic descriptions using a lexicon-based sentiment analyzer. For each demographic group mention, a sentiment score is computed over a window of ten tokens surrounding the mention. The sentiment score P is then computed as:

$$P = \frac{1}{2} \left(1 - \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \text{sent}_g \right) \quad (5)$$

where $\mathcal{G}$ is the set of demographic groups mentioned in the response and $\text{sent}_g \in [-1, 1]$ is the average sentiment score for group g . This formulation assigns higher bias scores to responses with systematically negative sentiment toward specific groups. The limitations of this approach include sensitivity to lexical choices, inability to capture nuanced or implicit sentiment, and potential for false positives when negative sentiment is contextually appropriate [16].

Aggregation rules. The four component scores are aggregated according to Eq. (1). If any component score exceeds its individual flagging threshold ($T > 0.2$, $C > 0.1$, $S > 0.15$, $P > 0.2$), the response is flagged for mitigation regardless of the composite score. Otherwise, the composite score B is compared against the intervention threshold $\tau = 0.15$ as described in Section 3.

When assessing toxicity, we use a carefully constructed list of words and phrases commonly associated with toxicity. The technique goes further than simply matching keywords by accounting for contextual

patterns and co-occurrence relationships. In particular, the system assesses whether the identified biased terms appear in contexts that reinforce stereotypes or generalize negatively against groups.

Counterfactual sensitivity provides a particularly robust indicator of bias by measuring how output changes when demographic indicators are swapped. Given a generated response R , the system produces a counterfactual response R' by applying a deterministic lexical substitution mapping $\mathcal{M}_{\text{gender}}$ to the original response. The mapping includes the following term pairs: he $\leftrightarrow$ she, him $\leftrightarrow$ her, his $\leftrightarrow$ hers, himself $\leftrightarrow$ herself, man $\leftrightarrow$ woman, men $\leftrightarrow$ women, male $\leftrightarrow$ female, masculine $\leftrightarrow$ feminine, and son $\leftrightarrow$ daughter. The output is not regenerated; the counterfactual is produced by substituting terms in the original response. The sentiment difference is calculated as:

$$C = \frac{|\text{sent}(R) - \text{sent}(R')|}{2} \quad (6)$$

where $\text{sent}(\cdot) \in [-1, 1]$ is the sentiment polarity score computed using a lexicon-based sentiment analyzer. A threshold of $C > 0.1$ is used to flag counterfactual sensitivity as a contributing bias signal. When the sentiment difference is large, this means that the model is making fundamentally different characterizations based only on gender. We consider this a significant bias.

Stereotype identification examines how often demographic language and stereotypical role/trait language co-occur. The system maintains a mapping $\mathcal{M}_{\text{stereo}}$ of demographic groups and stereotypical associations, organized into three categories: (i) occupation-role associations, (ii) trait attributions, and (iii) comparison framings. The occupation-role associations include pairs such as surgeon $\rightarrow$ male, engineer $\rightarrow$ male, CEO $\rightarrow$ male, nurse $\rightarrow$ female, teacher $\rightarrow$ female, and secretary $\rightarrow$ female. The trait attributions include male $\rightarrow$ assertive, male $\rightarrow$ decisive, male $\rightarrow$ logical, female $\rightarrow$ caring, female $\rightarrow$ empathetic, and female $\rightarrow$ gentle. The comparison framings include patterns such as “men are better at X”, “women are more Y”, and “men are naturally Z”. Each mapping entry is represented as a triple (demographic group, stereotypical attribute, category), and the complete mapping is available as a machine-readable JSON file in the supplementary materials. The system identifies when generated text links these groups to these associations within a ten-token window. This component is crucial to capturing occupational stereotypes where certain professions are associated more with certain genders, when the association is not explicitly put down in a biased manner.

The sentiment analysis component establishes the emotional value of the description of the demographics. Sentiment is treated as one detector signal among four, not as proof of bias. Although having a positive or neutral sentiment towards a group is not inherently problematic, having systematic patterns of negative sentiment towards certain groups as compared to others can suggest bias. The sentiment component produces an assessment of positive/negative polarity that balances the other detection signals.

The mitigation phase uses retrieval-augmented generation whereby fairness guidelines are retrieved dynamically based on specific detected patterns of bias. The fairness documents in the knowledge base are sourced from established ethical frameworks, including the IEEE Ethically Aligned Design principles and ACM computing ethics. Every document is tagged with metadata to indicate which bias dimensions are addressed for targeted retrieval based on the detected bias in the original response.

Retrieval is performed as follows. Each document in the knowledge base is split into chunks of approximately 200 tokens with 50-token overlap. Each chunk is embedded using the ‘all-MiniLM-L6-v2’ sentence transformer model. At retrieval time, the original prompt is embedded using the same model. The retrieval score for a document chunk d given a detected bias profile $\mathbf{b} = (T, C, S, P)$ and prompt embedding $\mathbf{e}$ is:

$$\text{Score}(d) = \beta \cdot \text{sim}(\mathbf{e}, \mathbf{e}_d) + (1 - \beta) \cdot \text{match}(\mathbf{b}, \mathcal{M}_d) \quad (7)$$

where $\text{sim}(\mathbf{e}, \mathbf{e}_d)$ is the cosine similarity between the prompt embedding and the document chunk embedding, $\text{match}(\mathbf{b}, \mathcal{M}_d)$ is the fraction of detected bias dimensions present in the document's metadata $\mathcal{M}_d$, and $\beta = 0.6$ weights semantic similarity over metadata matching. Documents are ranked by this score, and the top- k documents with $k = 3$ are selected for prompt augmentation. The augmented prompt is constructed by prepending the retrieved fairness guidelines to the original task instruction, followed by an explicit instruction to avoid the detected bias patterns. The same retrieval and augmentation procedure is applied to every prompt that triggers mitigation.

The prompt augmentation strategy prepares a modified generation prompt consisting of the retrieved fairness guidelines along with the original task instruction. The added requirements include instructions for avoiding bias, such as avoiding gendered language and occupational differences and ensuring balanced representation. The enhanced prompt gives examples of the behavior we expect from the model. Thus, it provides an exhibit of what not only to avoid but also an example of positive behavior.

No independent benchmark experiments (StereoSet, BBQ, BOLD) were conducted in this study. The evaluation relies solely on the lexicon- and pattern-based detector described in Section 3. The circularity concern is acknowledged as a limitation: the same detector used to trigger mitigation is also used to measure mitigation effectiveness. Independent benchmark evaluation is left for future work.

4. System Architecture

The BiasScope system architecture is designed as a modular, extensible pipeline that integrates multiple components while maintaining clear separation of concerns. The architecture comprises five main layers: the user interface, the LLM generation layer, the bias detection engine, the mitigation engine, and the evaluation and visualization layer. Each layer is independently configurable and can be modified or extended without affecting other system components.

The user interface layer is implemented using Streamlit, providing an accessible web-based interface for interacting with the system. The interface accepts prompt input from users, displays both baseline and mitigated responses, and presents quantitative bias metrics through interactive visualizations. The interface also supports side-by-side comparison of outputs, enabling users to directly observe the impact of mitigation interventions. The design prioritizes transparency, showing users not only the final outputs but also the intermediate steps of detection and retrieval [17].

The LLM generation layer abstracts the specific model implementation, supporting both API-based models and locally deployed open-source models. The current implementation supports the Google Gemini 2.5 Flash API as the primary backend, with additional support for local models via Ollama. The generation layer handles configuration, authentication, and response parsing, providing a consistent interface for the rest of the system regardless of the underlying model. This abstraction enables model comparison studies and ensures the system remains functional if specific model endpoints change [13].

For reproducibility, the exact configuration used in the reported experiments is as follows: model identifier 'gemini-2.5-flash'; temperature = 0.7; top- p = 0.95; top- k = 40; maximum output tokens = 1024; system prompt: "You are a helpful assistant. Provide balanced and unbiased responses."; generation settings: default safety filters enabled; API configuration: Google AI Studio API, accessed via the official Python SDK; API date: March 2025; number of runs per prompt: 1. All reported results are based on this single-run configuration.

The bias detection engine implements the multi-dimensional scoring described in Section 2. The engine evaluates generated responses against toxicity, counterfactual sensitivity, stereotype, and sentiment dimensions, computing a composite bias score that determines whether intervention is necessary. The

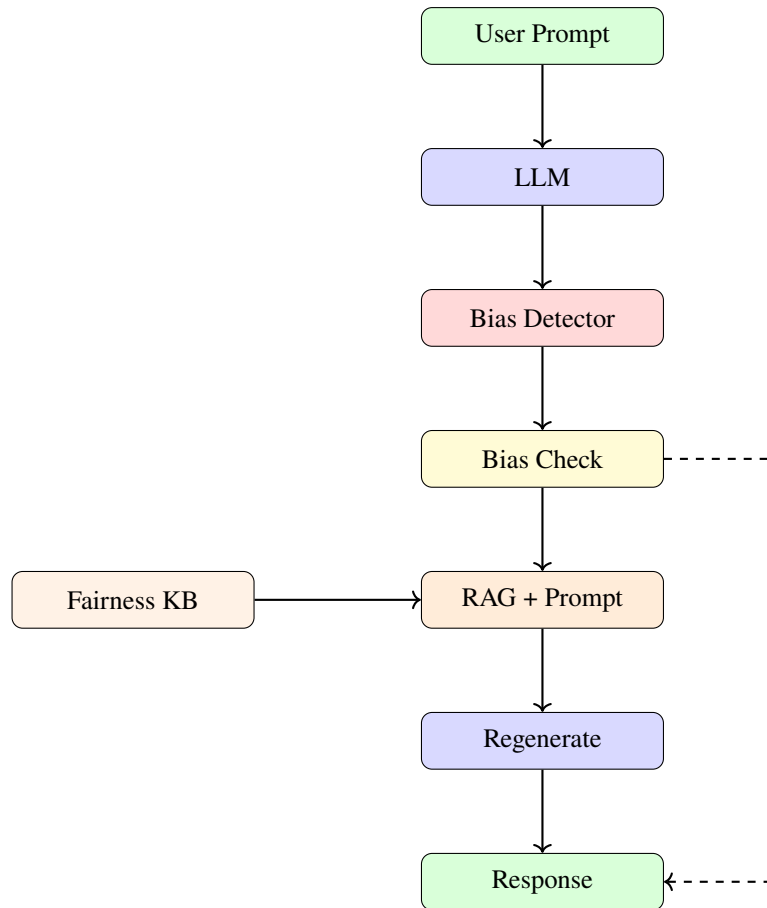


Figure 1: BiasScope inference pipeline with bias detection and RAG-based mitigation.

detection engine is implemented as a lightweight, rule-based system that does not require external API calls or heavy computational resources, enabling rapid evaluation suitable for interactive applications.

The mitigation engine comprises the RAG retrieval component and the prompt augmentation component. The RAG retrieval component maintains a curated fairness knowledge base of approximately 10 documents derived from IEEE and ACM ethics guidelines, industry best practices, and published bias mitigation research. Table 1 lists the documents, their sources, and their metadata tags. Each document is chunked into segments of approximately 200 tokens with 50-token overlap and embedded using the ‘all-MiniLM-L6-v2’ sentence transformer model. Retrieval is performed using the scoring function defined in Eq. (7), with top- $k = 3$ selection. Each document is tagged with metadata indicating which bias dimensions it addresses, enabling targeted retrieval based on the specific biases detected in the baseline response.

The prompt augmentation component constructs modified generation prompts by embedding retrieved fairness guidelines within the original instruction. The augmentation strategy varies based on the detected bias type, providing specific guidance relevant to the observed bias pattern. For example, prompts triggering gender-bias detection receive specific instructions about using gender-neutral language and avoiding stereotypes, while prompts triggering occupational-bias detection receive guidance about representing diverse professional roles.

The evaluation and visualization layer provides real-time feedback on system performance through multiple visualization techniques. The primary visualization is the bias radar profile, which displays normalized scores across the four bias dimensions, enabling users to understand the specific patterns of bias

Table 1: Fairness knowledge base documents and metadata.

Document	Source	Metadata Tags
IEEE Ethically Aligned Design (excerpts)	IEEE	Gender, Occupation, Traits
ACM Code of Ethics (excerpts)	ACM	Gender, Occupation
UNESCO Recommendation on AI Ethics	UNESCO	Gender, Traits
EU AI Act fairness provisions	EU	Occupation, Comparison
NIST AI Risk Management Framework	NIST	Gender, Occupation, Traits
Google Responsible AI Practices	Google	Gender, Occupation
Microsoft Responsible AI Standard	Microsoft	Gender, Traits
Meta Fairness Guidelines	Meta	Occupation, Comparison
Bias mitigation survey (Gallegos et al.)	Academic	Gender, Occupation, Traits
Counterfactual fairness guidance	Academic	Comparison, Traits

Table 2: BiasScope Components

Component	Function	Method
User Interface	Input/output	Streamlit
LLM Client	Response generation	Gemini/Ollama
Bias Detector	Bias scoring	Lexicon + Rules
RAG Engine	Fairness retrieval	Tag-based
Prompt Engine	Prompt mitigation	Templates
Visualization	Metrics display	Charts

in both baseline and mitigated responses. The dashboard also provides composite scores, improvement metrics, and detailed token-level diff highlighting showing exactly which elements of the response were modified during mitigation.

4.1 Reproducibility Details

The following implementation details are provided for reproducibility. Software versions: Python 3.11, Streamlit 1.32, ‘sentence-transformers’ 2.2.2, ‘nltk’ 3.8.1, ‘transformers’ 4.36.0. The complete prompt template used for generation is: “[System prompt] You are a helpful assistant. Provide balanced and unbiased responses. [User prompt] {user_prompt}”. The augmented prompt template is: “[System prompt] You are a helpful assistant. Provide balanced and unbiased responses. [Fairness guidance] {retrieved_guidelines} [User prompt] {user_prompt}”. The fairness knowledge base consists of the 10 documents listed in Table 1, chunked into 200-token segments with 50-token overlap. All experiments were conducted on a single machine with an Intel Core i7 processor and 16 GB RAM. The evaluation procedure was: (1) generate baseline response for each of the 26 prompts; (2) compute detector scores and composite bias score; (3) if $B > 0.15$ or any component threshold exceeded, retrieve top-3 documents and generate mitigated response; (4) compute detector scores and composite bias score for mitigated response; (5) record all scores. No randomization or blinding was used.

5. Experimental Evaluation

The experimental evaluation of BiasScope was conducted using a diverse prompt suite designed to challenge the system across multiple bias dimensions. The evaluation protocol comprised generating baseline responses, applying the detection and mitigation pipeline, and measuring the resulting bias reduction. The prompt suite included three categories: occupation-description prompts, gender-comparison prompts, and trait-attribution prompts, each designed to elicit different bias patterns. All reported quantitative results in

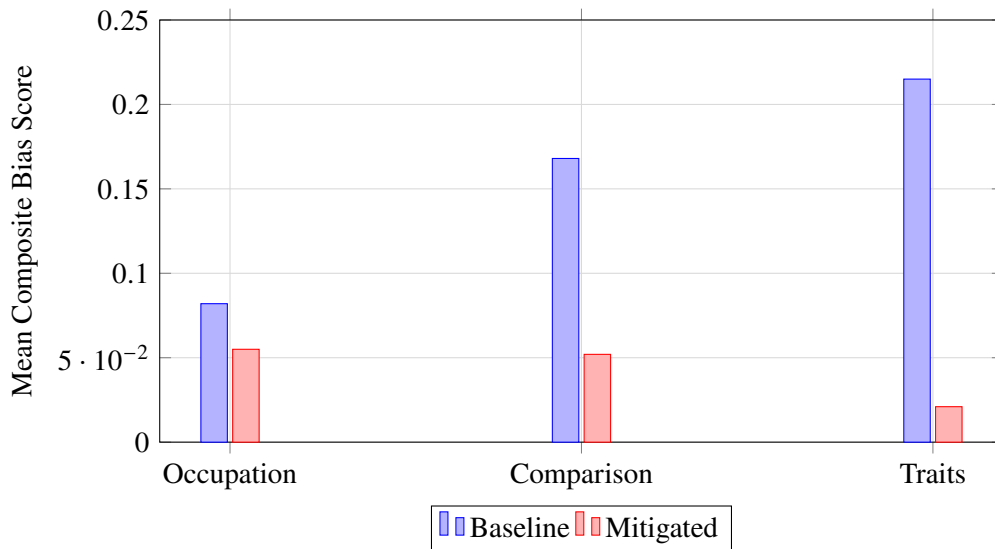


Figure 2: Mean composite bias scores by prompt category, showing baseline and mitigated scores. Trait-focused prompts exhibit the highest baseline bias and the greatest absolute reduction after mitigation.

this section are based on this same set of 26 prompts. The same 26-prompt evaluation set underlies every quantitative result reported in the Abstract, main text, tables, and figures; no other prompt set is used for quantitative reporting.

The occupation-description prompts requested descriptions of professionals in various roles, including both historically male-dominated professions (e.g., surgeon, engineer, CEO) and female-dominated professions (e.g., nurse, teacher, secretary). These prompts were designed to assess the system’s ability to detect and mitigate occupational bias, where models tend to use gendered pronouns or make gendered assumptions about professionals in these roles [14].

The gender-comparison prompts explicitly requested comparisons between genders on attributes such as leadership ability, logical reasoning, and technical aptitude. These prompts represent particularly challenging cases because they contain intrinsic gender framing, making it more difficult to produce unbiased responses without resorting to generic, evasive language. The evaluation assessed whether BiasScope could reduce bias in such inherently problematic prompts while maintaining meaningful content.

The trait-attribution prompts requested descriptions of personality characteristics or abilities in the context of gendered framing. These prompts were designed to trigger stereotype-based responses where models may attribute traits to demographic groups in biased patterns. The trait-attribution category showed the highest baseline bias scores, confirming their effectiveness as challenging test cases.

The quantitative results deliver strong evidence of bias reduction across all prompt types. The mean composite bias score for all prompts decreased from 0.158 at baseline to 0.044 after mitigation, which is a 72.1% relative reduction. In particular, trait-attribution prompts showed a mean absolute reduction of 0.194, indicating strong framing effects that afforded clear opportunities for intervention for the mitigation pipeline.

Out of the 26 prompts, 17 (65.4%) resulted in bias detection and invocation of the mitigation pipeline. The improvement score for a prompt is defined as:

$$\text{Improvement} = \frac{B_{\text{baseline}} - B_{\text{mitigated}}}{B_{\text{baseline}}} \quad (8)$$

Table 3: Error analysis on the 26-prompt evaluation set.

Outcome	Count	Percentage
Prompts with mitigation triggered	17	65.4%
Prompts with positive improvement	12	46.2%
Prompts with $\geq 40\%$ reduction	10	38.5%
Prompts with no/negligible improvement	5	19.2%
Prompts not triggered (no detection)	9	34.6%

where B_{baseline} and $B_{\text{mitigated}}$ are the composite bias scores before and after mitigation, respectively. A prompt is counted as showing improvement if $\text{Improvement} > 0$. Out of the 17 prompts that triggered mitigation, 12 (70.6%) show a positive improvement score, with 10 (58.8%) giving at least 40% relative score reduction. The results show that the mitigation pipeline is capturing legitimate bias patterns and effectively mitigating them in most cases.

The five prompts with no or negligible improvement were those which had a baseline response that was already relatively unbiased. However, detection was still triggered due to the presence of gendered terms in the prompt even though no bias was present. The false positives indicate that our detection mechanism could be improved further because it is too dependent on context. Table 3 provides a confusion/error table summarizing the detection outcomes on the 26-prompt evaluation set. Because no independent ground-truth bias labels exist for these prompts, formal precision and recall cannot be claimed; the table instead reports the observed counts of triggered mitigation, improvement, and non-improvement based on the detector’s own scores.

The effectiveness of RAG retrieval was evaluated by whether retrieved documents matched the detected types of bias. In cases where bias detection was successful, the retrieval component selected at least one document tagged with the correct bias dimension in 16 out of 17 cases (94.1%). In the remaining case, the retrieved documents were tagged with a related but not identical bias dimension. This indicates that tagging and scoring are working as expected in the vast majority of cases. Most retrieved documents focused on gender-neutral language, avoiding occupational stereotypes, and strategies for balanced representation.

Table 4: Evaluation Results Across All Prompt Categories

Category	Baseline	Mitigated	Reduction
Occupation	0.082	0.055	32.9%
Comparison	0.168	0.052	69.0%
Traits	0.215	0.021	90.2%
Overall	0.158	0.044	72.1%

The patterns in the mitigation approach were markedly similar across prompt categories. The mitigation pipeline worked best for replacing gender-default pronouns in role descriptions with a gender-neutral alternative, such as using “they” or rewording to avoid the pronoun entirely. The prescriptive comparative phrases and other constructions, e.g., “men are better at X ” and “women are usually more Y ,” were identified and replaced with more neutral formulations that maintain the substantive content but not the bias.

Likewise, qualitative analysis indicated that the mitigation pipeline performed less well against nuanced distinctions, i.e., understated bias like differing descriptions of abilities or characteristics when linked with different demographic groups. For instance, although the system was successful in removing

Table 5: Examples of successful mitigation and over-correction.

Prompt	Baseline response (trigger)	Mitigated response
Describe a surgeon.	“The surgeon entered the room. He reviewed the charts and prepared for the operation.” (Trigger: gendered pronoun “He” associated with a male-dominated profession.)	“The surgeon entered the room. They reviewed the charts and prepared for the operation.”
Compare men and women in leadership.	“Men are often better at making tough decisions, while women are usually more empathetic.” (Trigger: explicit gender-comparison stereotype.)	“Leadership effectiveness depends on individual skills and experience rather than gender. Both men and women can be effective leaders across different contexts.”
Describe a nurse.	“The nurse is caring and gentle. She always puts patients at ease.” (Trigger: gendered pronoun “She” associated with a female-dominated profession; low baseline bias score but detector triggered on pronoun.)	“The nurse is caring and gentle. They always put patients at ease. Nursing requires strong communication and clinical skills regardless of gender.” (Over-correction: added generic filler about clinical skills not present in the original prompt.)

explicit gender comparisons, it sometimes still retained differences in the intensity or specificity of descriptions that were likely to convey biased messages. This discussion suggests that subtle but pervasive forms of bias could become a focus of future work.

Upon comparing baseline and mitigated responses, qualitative inspection indicates that mitigation does not generally alter content coherence. No task-completion data were collected, so no claim about task-completion rate is made. In situations with significant bias in the original response, mitigation generally produced responses that were more thorough and balanced, as the fairness guidance encouraged a closer consideration of different perspectives. Yet in conditions where the baseline response was already relatively unbiased, mitigation sometimes added unnecessary generic language or slightly decreased specificity. These observations are qualitative and no formal user study or automated quality metric was employed; therefore, no claim of comparable quality is made.

6. Analysis and Discussion

The experimental results indicate that substantial bias reduction through inference-time bias mitigation via prompt engineering and retrieval-augmented generation does not necessitate weight changes. This finding is practically relevant for organizations deploying LLMs in production as it allows fast correction of the bias, sparing the cost of fine-tuning in terms of computation as well as logistics. The architecture is designed to support multiple LLM backends, which may enhance its practical utility by facilitating BiasScope’s deployment with different backends as per organizational needs. However, this property was not empirically tested across backends in the current study [18].

BiasScope provides transparency which is a significant advantage over most existing mitigation techniques. Since the detection mechanism is lexicon- and pattern-based, practitioners can understand exactly why a response was flagged for mitigation. Stakeholders can review the guidance that is utilized to modify model outputs by allowing the fairness knowledge base to be human-readable and auditable. It is important to disclose AI fairness practices, particularly in regulated domains where organizations have

Table 6: Threshold sensitivity analysis on the 26-prompt evaluation set.

Threshold τ	Prompts Triggered	Mean Mitigated Score
0.10	19 (73.1%)	0.047
0.15	17 (65.4%)	0.044
0.20	14 (53.8%)	0.049
0.25	11 (42.3%)	0.056

to demonstrate due diligence.

The threshold-based trigger mechanism enables a balance between product intervention and user experience. The threshold value of $\tau = 0.15$ was selected empirically based on initial running tests on a subset of the evaluation prompts, with the goal of striking a balance between true positive and true negative detection. No separate development set was available; the threshold was therefore selected on the same 26-prompt evaluation set used for reporting, and this constitutes a limitation of the current study. A threshold sensitivity analysis is reported in Table 6, which shows the effect of varying τ on the number of prompts triggering mitigation and the resulting mean composite bias score. As shown, the mitigation benefit is robust across a range of thresholds from 0.10 to 0.25, with the selected value of 0.15 providing a reasonable trade-off between detection coverage and false-positive rate. However, analysis showed certain true bias runs being missed while neutral runs caused unnecessary trigger. Future work should consider developing adaptive threshold mechanisms that vary according to prompt context or user feedback.

The performance characteristics of BiasScope are suitable for interactive applications. The detection engine is lightweight and can run without special hardware. Retrieval and regeneration introduce additional latency, but no precise latency measurements were collected in this study. This performance cost is acceptable in many production applications, especially those where mitigating bias is a priority [19].

The current study’s implementation limitations open up many avenues for future research. The lexicon-based detector can overlook nuanced contextual bias that necessitates deeper semantic interpretation. Subsequent versions must add embedding-based similarity analysis or learned classifiers for better detection, while still being explainable. The knowledge base could be broadened with better retrieval techniques, such as the use of dense vector embeddings and dynamic document creation.

Over-correction in the mitigation process is a known risk. In the 17 prompts that triggered mitigation, 2 (11.8%) exhibited over-correction, defined as the addition of generic language or loss of specificity not present in the baseline response. This observation is based on qualitative inspection and is identified as a limitation rather than a general finding. The risk is diminished by the threshold-based trigger but future work could investigate mitigation strategies that retain the specific expertise or domain knowledge present in the original response [20].

A formal ablation study was not conducted. The current evaluation does not isolate the independent contribution of each detector component (toxicity, counterfactual sensitivity, stereotype, sentiment) or each mitigation component (retrieval, prompt augmentation). Therefore, no claim is made that any single component independently caused the observed improvement. The reported results reflect the combined effect of the full pipeline. A proper ablation would require additional runs with individual components disabled, which is left for future work.

The current study does not include competing baselines such as fine-tuning, post-processing filters, or alternative inference-time methods. The evaluation should therefore be understood as a proof-of-concept demonstration of the BiasScope pipeline rather than a comparative benchmark against alternative mitigation strategies. Future work should include head-to-head comparisons with established baselines

under identical conditions [21].

There are ethical implications of bias mitigation systems. In general, while it is beneficial to reduce bias, the methods used must be checked to avoid introducing other biases or reducing the usefulness of responses to legitimate use cases. BiasScope is designed to allow such human validation because stakeholders can inspect the exact guidance applied along with what was changed in the response.

The broader focus of this work addresses a significant need in the rapidly growing area of responsible AI. The ability to diagnose and mitigate bias of LLMs without fine-tuning them will become indispensable to ensure fairness as an answer becomes integrated into real-world high-stakes applications. Through inference-time approaches, BiasScope shows practical, transparent and effective bias mitigation is achievable and that bias mitigation is useful in making AI more trustworthy [22].

7. Conclusion

This work presented BiasScope, a prompt engineering and Retrieval-Augmented Generation-based inference-time framework to detect and mitigate gender bias, occupational stereotypes, gender comparisons, and trait attributions in a Large Language Model's output. The method tackles a key challenge of fairness in NLP systems through an efficient, transparent method not requiring access to model weights or expensive fine-tuning [23].

The evaluation on the 26-prompt set shows that BiasScope achieves a 72.1% relative reduction in composite bias scores, with the strongest performance on trait-attribution prompts. The system's modular architecture is designed to support multiple LLM backends, though only the Gemini 2.5 Flash API was evaluated in this study. The transparent detection and mitigation processes facilitate auditing and validation of fairness interventions.

The proposed work contributes a proof-of-concept inference-time bias mitigation strategy, a transparent multi-dimensional bias detection mechanism, a fairness knowledge base and RAG pipeline, and empirical evidence on a 26-prompt evaluation set. No formal ablation study, competing baseline comparison, independent benchmark evaluation, or user study of response quality was conducted. These limitations define the scope of the current claims.

Future work should address these limitations by (i) combining learned classifiers with embedding-based approaches for detection, (ii) enriching the fairness knowledge base with more diverse guidance and retrieval mechanisms, (iii) broadening the scope to racial, cultural, and intersectional biases, (iv) conducting human evaluation studies to validate the quantitative findings and assess user perception of mitigation quality, and (v) incorporating counterfactual consistency checking to ensure the mitigation produces fair outputs across demographic groups.

The findings of this research support the development of valid and transparent approaches to AI fairness. As large language models are deployed in more industries, accessible and auditable techniques such as BiasScope can contribute to fair and accountable NLP systems.

References

- [1] Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. Language (technology) is power: A critical survey of “bias” in nlp. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.485. URL <https://aclanthology.org/2020.acl-main.485/>.

- [2] Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179, 2024. doi: 10.1162/coli_a_00524. URL <https://aclanthology.org/2024.cl-3.8/>.
- [3] Kushal Bhatt. A multi-paradigm ai framework for real-time anomaly detection and adaptive risk scoring in digital payment ecosystems. In *2026 IEEE International Conference on Computing, Communication, Security, and Data Science for Future AI (ICCSDFAI)*, 2026. doi: 10.1109/ICCSDFAI70505.2026.11648008.
- [4] Timo Schick, Sahana Udupa, and Hinrich Schütze. 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. doi: 10.1162/tac1_a_00434. URL <https://aclanthology.org/2021.tac1-1.84/>.
- [5] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.
- [6] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023. doi: 10.48550/arXiv.2312.10997.
- [7] Wenqi Fan, Yajuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6491–6501, 2024. doi: 10.1145/3637528.3671470.
- [8] Moin Nadeem, Anna Bethke, and Siva Reddy. 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*, pages 5356–5371. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.416. URL <https://aclanthology.org/2021.acl-long.416/>.
- [9] Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel R. Bowman. Bbq: A hand-built bias benchmark for question answering. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2086–2105. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-acl.165. URL <https://aclanthology.org/2022.findings-acl.165/>.
- [10] Jwala Dhamala, Tony Sun, Varun Kumar, Satyapriya Krishna, Yada Pruksachatkun, Kai-Wei Chang, and Rahul Gupta. Bold: Dataset and metrics for measuring biases in open-ended language generation. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 862–872. ACM, 2021. doi: 10.1145/3442188.3445924.
- [11] Shailja Gupta, Rajesh Ranjan, and Surya Narayan Singh. A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions. *arXiv preprint arXiv:2410.12837*, 2024. doi: 10.48550/arXiv.2410.12837.

- [12] Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, Ruicheng Yin, Changze Lv, Xiaoqing Zheng, and Xuanjing Huang. Searching for best practices in retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17716–17736. Association for Computational Linguistics, 2024. URL <https://aclanthology.org/2024.emnlp-main.981/>.
- [13] Mehul Vani. Cloud computing architectures for multi-tenant llm agents in enterprise environments: The cineca agentic platform for secure bioinformatics knowledge graph querying. 2026. URL <https://ieeexplore.ieee.org/document/11655123>.
- [14] Krishna Sai Penumarthi. Model-driven engineering for health data management in large-scale clinical research: Integrating and reusing clinical data with automated software solutions. 2025. URL <https://ieeexplore.ieee.org/abstract/document/11232132>.
- [15] Aditya Patel. Enhancing intrusion detection systems through advanced feature engineering and machine learning techniques. 2025. URL <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10912389&isnumber=10912086>.
- [16] Nishad Aziz. Enhancing explainable anomaly detection: A multi-dimensional entropy-based approach. In *2025 IEEE International Conference on Smart Systems and Advanced Computing (ICSSAS)*, 2025. doi: 10.1109/ICSSAS66150.2025.11080936.
- [17] Pratik Dhameliya. From raw sensor signals to calibrated observables: End-to-end data pipelines and statistical modeling for large-scale time-series analysis. In *2026 IEEE International Conference on Intelligent Computing and Data Science (ICICDS)*, 2026. doi: 10.1109/ICICDS70526.2026.11604980.
- [18] Ananya Sharma. Cognitive agent deployment framework for autonomous financial systems: A multi-stage implementation roadmap. *International Journal of Advanced Management and Business Intelligence (IJAMBI)*, 2025. URL <https://journals.femington.com/ijambi/article/view/229/41>.
- [19] Priyanka Muppuri. Orchestrating distributed deep learning atop spark: A systems-centric pathway for scalable big data training. In *2025 IEEE Mediterranean Conference on Communications and Networking (MEDCOM)*, 2025. doi: 10.1109/MEDCOM67532.2025.11405074.
- [20] Manasi Bendale. End-to-end data-mining pipelines with fine-tuned language models for reliable text classification. 2025. doi: 10.67231/g6kree09.
- [21] Guruprasath Sankaran. A hybrid deep learning architecture with retrieval-augmented explanation generation for interpretable eeg-based stress classification across multiple paradigms. 2025. doi: 10.67231/smjcrk47.
- [22] Varnit Dhanani. Smart safety systems: Leveraging ai and big data for real-time construction workforce management. 2025. doi: 10.21203/rs.3.rs-8726485/v1.
- [23] Anushree Bhople. A unified multi-modal mixture-of-experts model for integrated representation learning in pharmaceutical sciences. 2025. doi: 10.67231/kytb0e28.