Navin Manaswi[1*]

Inference Functions in Large Language Models: A Comprehensive Framework for Bias Mitigation

Journal of Electrical Systems

Abstract
Large Language Models (LLMs) have become a cornerstone of natural language processing tasks across industries. However, these models often perpetuate the biases present in the training data, resulting in harmful societal impacts. Bias in LLMs can manifest in various forms, such as gender stereotypes, racial prejudice, and cultural misrepresentation. This paper introduces **Inference Functions**, a post-processing mechanism designed to dynamically detect and mitigate bias in real-time during the inference stage. Unlike traditional bias mitigation techniques, which require pre-processing data or retraining models, inference functions offer a scalable and efficient solution by intervening after the model generates outputs. We explore the design, application, and trade-offs of inference functions for bias mitigation, backed by experiments and case studies. We also discuss the ethical implications and potential for compliance with emerging AI regulations.

Keywords: racial prejudice, mitigation, Large Language Models (LLMs), misrepresentation

1. INTRODUCTION

As Large Language Models (LLMs) grow in size and sophistication, their outputs increasingly influence critical decision-making in verticals such as healthcare, legal services, education, and hiring. Despite their impressive capabilities, LLMs often reflect the biases and toxicity present in their training data, which are derived from large-scale internet corpora that contain societal prejudices. These biases can propagate gender stereotypes, racial prejudices, and cultural misrepresentations, leading to unethical and potentially harmful outcomes.

Let's go through its mathematical definition. While a precise mathematical definition of an inference function can vary depending on its specific implementation, we can define it generally.
Let:

- `Y` represents the LLM outputs. It could be strings, lists of tokens, or any other LLM output
- `Y'` represents the *modified* outputs after the inference function is applied.

Then, the inference function `f` is a mapping: `f: Y -> Y'`
It means that `f` takes an element from the set of possible LLM outputs (`Y`) and transforms it into an element in the set of modified outputs (`Y'`).
Examples:

- **Simple substitution:** If `Y` is the set of all strings, `f` could be a function that replaces specific words or phrases with less biased alternatives. The function `f`can be mapping from a string `y` to a new string `y'` where certain substrings are replaced.
- **Sentiment adjustment:** If `Y` represents text with an associated sentiment score, `I` could adjust the output to achieve a more neutral sentiment. The function `f` could be a function that maps a (text, sentiment score) pair in `Y` to a new (text, adjusted sentiment score) pair in `Y'`.
- **Toxicity filtering:** If `Y` is the set of all text sequences, `f` could be a function that assigns a "toxicity score" to each sequence and removes or modifies those exceeding a certain threshold.

Note that the specific form of `f` will depend on the desired fairness and debiasing goals. `f` can be deterministic (always producing the same output for the same input) or probabilistic (introducing some randomness in the modification process).
As LLMs learned from real-world data, it might unintentionally include biased or unfair language in its responses. For example, it might make assumptions based on gender, race, or culture because that's what it saw in the data.
An **inference function** acts like a filter or a safety check that sits between the model and the user. After the model generates an answer, the inference function reviews it to see if there's any bias or unfairness in the response.
Here's how it works:

1. **The model generates an output**: Say you ask it a question, and it gives you an answer, but that answer might have biased language.
2. **The inference function checks for bias**: It scans the answer to see if any biased words, phrases, or patterns appear. For example, if the model says something like "nurses are always women," the inference function detects this as biased.
3. **Fixing the bias**: If the inference function detects bias, it changes the answer to make it fair. So instead of "nurses are always women," it might change it to "nurses can be men or women."
4. **No bias, no change**: If the answer is fair, the inference function lets it pass through without making any changes.

[1*]LLM Researcher, Ex-Google Developers Expert, Ex-Guest Faculty at IIT Kgp

So, for a regular person, this function is like a smart editor that automatically corrects any unfair or biased language in real-time, making sure the AI's responses are ethical and neutral before you see them.

Here are some of the typical inference functions utilized to enhance LLM safety:

1. Bias Detection and Correction

- **Rule-Based Inference**: This approach relies on predefined rules to identify and correct biased or inappropriate content. For instance, gender-specific pronouns such as “he” or “she” can be replaced with the neutral term “they” to avoid reinforcing stereotypes.
- **Bias Classifiers**: These classifiers are designed to detect biases based on gender, race, or culture within the output. Once identified, these biases are corrected to ensure fairness across demographic groups.
- **Counterfactual Generation**: By swapping demographic attributes such as gender or race, the system checks whether the generated response changes. If it does, this suggests bias, which is then corrected.
- **Fairness Constraints**: These constraints, such as demographic parity or equalized odds, ensure that model outputs do not disproportionately favor one group over another. When biases are detected, outputs are adjusted accordingly.

2. Toxicity and Harmful Content Detection

- **Toxicity Filters**: These filters detect and remove harmful language, including hate speech, threats, or harmful stereotypes. If such content is identified, the inference function either blocks it or rewrites the response.
- **Sentiment Analysis Modifications**: When an output is overly negative or promotes harmful attitudes, sentiment analysis can neutralize the content. This ensures responses are balanced and not harmful.

3. Fairness-Aware Language Models

- **Word Embedding Post-Processing**: After the model generates an output, post-processing can adjust word embeddings to minimize bias. For instance, “hard debiasing” removes biased dimensions from word representations.
- **Equality-Constrained Inference**: This function ensures that outputs conform to fairness guidelines by preventing biased token sequences from being selected.

4. Decoding Strategy Modifications

- **Top-k or Top-p Sampling Adjustments**: These sampling techniques can be adjusted to reduce the likelihood of harmful or biased words being included in the output. For instance, certain problematic tokens can be down-weighted or excluded.
- **Debiasing Through Re-Ranking**: When generating multiple outputs (e.g., beam search), inference functions can re-rank them based on safety or bias considerations, selecting the least biased option.

5. Output Sanitization and Redaction

- **Keyword Replacement**: Sensitive or offensive language is detected and replaced with neutral alternatives. Predefined keyword lists help in identifying problematic words or phrases.
- **Redaction Mechanism**: In contexts requiring confidentiality, such as legal or medical fields, inference functions redact sensitive information, such as names or personal details, ensuring privacy is protected.

6. Explainability and Transparency Functions

- **Explanation Augmentation**: In high-risk situations, outputs are accompanied by explanations to clarify how the model reached its conclusion, providing greater transparency and improving user trust.
- **Confidence Calibration**: When the model is uncertain about its prediction, the inference function reduces the confidence in the output or flags it for further review. This prevents overconfidence in potentially inaccurate or harmful outputs.

7. Ethical Alignment with Human Preferences

- **Human Preferences Alignment**: Human feedback can guide inference functions to align outputs with ethical standards. When an output conflicts with human values, the inference function modifies it to align with those preferences.
- **Moral Filters**: These filters ensure outputs comply with predefined ethical principles. For example, in educational applications, content that promotes unethical behaviour is filtered out.

8. Safety-RLHF (Reinforcement Learning from Human Feedback)

- This technique leverages feedback from human users to train reward models that favor safer and more ethical responses. During inference, the reward model guides the LLM to select outputs that maximize safety and minimize potential harm.

9. Privacy-Preserving Inference

- **Differential Privacy Mechanisms**: In cases where sensitive data is involved, differential privacy mechanisms add noise to the model’s output distribution. This helps prevent the inadvertent leakage of personal details.
- **Entity Redaction**: This mechanism automatically detects and redacts personally identifiable information (PII), such as names or addresses, to ensure privacy compliance.

10. Contextual Sensitivity Adjustments

- **Context-Aware Filtering**: The inference function tailors the output to the context of the user, adjusting content based on factors such as age or location. For example, an LLM designed for children would filter out adult themes.
- **Cultural Sensitivity Adjustments**: In a global context, inference functions can adjust the output to avoid culturally insensitive or inappropriate content based on the audience.

Traditionally, bias mitigation in LLMs has relied on techniques like **data augmentation**, **adversarial debiasing**, and **post-processing**. However, these methods are limited by their dependence on pre-existing data and training processes, making them resource-intensive and impractical for real-time systems. **Inference Functions**, in contrast, provide a dynamic solution by detecting and mitigating bias during the inference stage, after the model has generated its outputs. This real-time, post-hoc correction offers a flexible, non-intrusive alternative that does not require retraining or fine-tuning the core model.

In this paper, we explore inference functions in detail, comparing them with existing bias mitigation techniques and discussing their application, limitations, and broader ethical implications.

2. WHAT ARE INFERENCE FUNCTIONS IN THE CONTEXT OF LLMS?

Inference functions refer to post-processing operations applied after the LLM generates its outputs. These functions detect and correct biased or harmful content before presenting the final output to the user. The primary goal of inference functions is to ensure fairness, neutrality, and ethical alignment of LLM outputs without modifying the underlying model.

2.1 Rule-Based vs. Learned Inference Functions

Inference functions can operate in two distinct ways:

- **Rule-based systems** rely on predefined rules or keyword detection to flag and adjust biased language. For instance, a rule-based function could replace gender-specific pronouns ("he," "she") with neutral terms ("they") in outputs where gender is irrelevant.
- **Learned inference models** use machine learning techniques to detect bias patterns based on prior examples and modify outputs in a context-aware manner. These functions are more flexible and adaptable, capable of handling nuanced biases that rule-based systems might miss.

By focusing on the output stage, inference functions can be seamlessly integrated into real-time systems, allowing for dynamic, context-sensitive bias mitigation without the need for retraining the LLM.

3. EXISTING BIAS MITIGATION TECHNIQUES

Before diving into the capabilities of inference functions, it is important to review existing techniques commonly used for bias mitigation in LLMs. Each approach operates at different stages of the model lifecycle and comes with its own strengths and limitations.

3.1 Data Augmentation and Balancing

Data augmentation involves modifying or enhancing the training data to ensure a more balanced representation of different demographic groups. This can include adding more examples from underrepresented groups or rebalancing the dataset to reduce bias. While effective at addressing certain biases, this method cannot eliminate deeper systemic biases present in the data.

3.2 Adversarial Debiasing

Adversarial debiasing uses an adversarial network to detect and reduce bias during training. In this setup, the main model learns to generate accurate predictions, while the “bias detective” identifies the patterns that suggest unfairness or prejudices in LLM outputs. Although adversarial debiasing shows promise in reducing biases, it is computationally expensive and hard to generalize and can be difficult to fine-tune across diverse domains.

3.3 Post-Processing Techniques

Post-processing techniques adjust the outputs of LLMs after the inference stage by re-ranking or modifying the text based on fairness metrics. While efficient, these methods are limited in their ability to address the root causes of bias and can lead to superficial corrections that compromise the coherence or relevance of the text.

Limitations of Existing Methods

Most of these techniques either require retraining the model or are limited in their ability to handle biases in real-time. In contrast, inference functions provide a scalable, flexible, and dynamic solution by operating exclusively during the inference phase, making them ideal for real-world applications where retraining is impractical.

4. LEVERAGING INFERENCE FUNCTIONS FOR BIAS MITIGATION

Inference functions are designed to intervene during the model’s inference stage, providing real-time bias detection and correction. This section explores how inference functions operate and their applications in bias mitigation.

4.1 Bias Detection

Inference functions begin by detecting bias using a variety of techniques:

- **Keyword detection**: Predefined lists of sensitive terms are used to flag biased language. For example, terms that reinforce gender stereotypes (e.g., "pilot" as "he") can be identified and flagged for correction.
- **Fairness metrics**: These metrics assess whether outputs disproportionately favor one group over another. For instance, a job recommendation system might evaluate whether suggestions are biased toward one gender or race.

- **Bias classifiers**: Machine learning-based classifiers trained on labeled datasets can identify and flag biased patterns in outputs, such as negative sentiment associated with certain ethnic groups.

4.2 Output Adjustment

Once bias is detected, the inference function modifies the output to ensure fairness. Techniques for output adjustment include:

- **Bias substitution**: Biased terms or phrases are replaced with neutral alternatives, ensuring that outputs are gender-neutral, culturally sensitive, and free from harmful stereotypes.
- **Re-ranking and selection**: In cases where multiple candidate outputs are generated, inference functions can re-rank or select the least biased output.
- **Counterfactual generation**: This involves re-analyzing the output by swapping demographic attributes (e.g., gender, race) to check for bias. If a response changes significantly based on these swaps, the output is adjusted to remove bias.

4.3 Feedback Loops for Continuous Improvement

Inference functions can be continuously improved through feedback loops. Outputs flagged for bias can be logged and analyzed, contributing to retraining and improving the inference function's accuracy. Human-in-the-loop (HITL) systems, where users provide feedback on biased outputs, can further refine these functions.

5. EXPERIMENTAL DESIGN AND EVALUATION FRAMEWORK

Testing the effectiveness of inference functions requires a rigorous experimental design that evaluates both bias mitigation and performance trade-offs.

5.1 Dataset Selection

We propose using benchmark datasets known for their biases, such as the **Winogender Schemas** (for gender bias) and **Sentiment140** (for racial bias). These datasets will serve as the foundation for testing the ability of inference functions to mitigate bias.

5.2 Baseline Model Evaluation

Before applying inference functions, the performance of the baseline LLM should be evaluated on the selected datasets. This will involve tracking metrics such as:

- **Sentiment balance**: Analyze how different demographic groups are portrayed in terms of positive, neutral, or negative sentiment.
- **Representation diversity**: Evaluate the diversity of names, pronouns, and associations across different demographic groups.

5.3 Implementation of Inference Functions

Inference functions will be implemented on top of the baseline model to detect and mitigate bias in real-time. This experiment will compare two groups:

- **Baseline model**: No bias mitigation.
- **Model with inference functions**: Bias detection and output modification in real-time.

5.4 Metrics and Performance Evaluation

Performance metrics will include:

- **Bias-related metrics**: Fairness, diversity, and representation scores.
- **Traditional metrics**: Fluency, coherence, and accuracy, as measured by standardized evaluation techniques (e.g., BLEU, METEOR).
- **Human evaluation**: Human annotators will assess the quality and fairness of the outputs, with inter-annotator agreement used to validate subjective judgments.

6. LIMITATIONS AND CHALLENGES

Despite their advantages, inference functions come with limitations and challenges that must be addressed:

- **Latency**: Inference functions introduce additional computational overhead, potentially increasing response times. In high-speed applications (e.g., conversational agents), this latency could impact user experience.
- **Overcorrection**: In some cases, inference functions may overcorrect, altering outputs unnecessarily and reducing the fluency or coherence of the generated text.
- **Subtle Bias**: Inference functions may struggle to detect subtle biases, particularly those related to cultural or contextual nuances. This remains an area for further research and refinement.

7. REAL-WORLD CASE STUDIES WITH STEP-BY-STEP DESIGN

To illustrate the application of inference functions in practice, consider the following case studies:

7.1 Gender Bias Mitigation for a Job Recommendation System

In a job recommendation chatbot powered by a large language model (LLM), users may input job inquiries and receive responses that contain gender bias (e.g., suggesting specific roles like "receptionist" or "engineer" based on implied gender preferences). These biases could harm both user experience and fairness by reinforcing stereotypes.

To develop an **inference function** that detects and mitigate gender bias in the output and replaces biased terms with neutral alternatives to improve fairness.

Step-by-Step Design of the Inference Function

Step 1: Bias Detection Mechanism

1. **Identify Gender-Specific Terms**: The first part of the inference function is to detect terms that reflect gender biases. In the context of job recommendations, biased terms might include:
   - Gendered pronouns: "he," "she," "his," "her."
   - Job-specific gender assumptions: "nurse" (assumed to be female), "engineer" (assumed to be male), "secretary" (assumed to be female).
2. **Context-Aware Bias Detection**: The inference function can use pattern recognition or NLP techniques like **named entity recognition (NER)** to determine the job title and gender-related terms in the same context. For instance, if "nurse" is mentioned along with the word "woman," the system flags this as a potential bias.
3. **Sentiment and Polarity Check**: Use sentiment analysis to check whether biased or stereotypical terms are associated with either positive or negative sentiment. For example, associating "caring" only with female-dominated jobs might signal bias.

Step 2: Bias Mitigation Mechanism

1. **Substitute Gendered Terms with Neutral Terms**:
   - Replace gender-specific terms like "he" or "she" with neutral alternatives like "they."
   - Replace job titles that carry gender stereotypes with neutral phrases. For example, "female nurse" becomes "nurse," and "male engineer" becomes "engineer."
2. **Contextual Re-Ranking of Output**:
   - If the LLM generates multiple outputs, the inference function re-ranks them based on bias scores. The least biased version (according to the fairness metric) is chosen for display.
3. **Fairness Correction**: If bias is detected, the system rephrases the sentence to be more inclusive. For instance:
   - Before: "He is applying for a python developer position because men are great leaders."
   - After: "They are applying for a python developer position because leadership depends on skills, not gender."

After this inference function implementation, outputs are modified in bias mitigation direction if needed. For example-
Original: She is a great nurse because women are naturally caring.
Mitigated: They are a great nurse because people are naturally caring.

7.2 Legal Document Generation

In the legal field, biased language related to race or ethnicity can have serious consequences. For example, an LLM used to draft legal documents may inadvertently use racially charged language when describing specific cases. Inference functions tailored to this domain can detect and remove such language, ensuring that outputs are legally and ethically compliant. Similar steps are taken for bias mitigation.

8. BROADER IMPLICATIONS FOR RESPONSIBLE AI AND ETHICAL CONSIDERATIONS

The use of inference functions extends beyond bias mitigation to promote broader principles of responsible AI:

- **Fairness and Inclusivity**: Inference functions enable LLMs to produce outputs that are inclusive and fair, reducing the risk of discrimination.
- **Ethical Accountability**: By reducing bias in real-time, inference functions promote ethical AI systems that align with societal values and norms.
- **Regulatory Compliance**: As AI governance frameworks such as the **EU AI Act** and **U.S. AI ethics guidelines** evolve, inference functions provide a practical solution for ensuring compliance with fairness and non-discrimination regulations.
- **Building Trust**: By consistently generating fair, unbiased outputs, inference functions increase user trust, which is critical for the long-term success of AI systems in sensitive domains.

9. CONCLUSION

Inference functions offer a scalable and dynamic solution to the challenge of bias in LLMs. By providing real-time bias detection and correction, they ensure that LLM outputs align with ethical and fairness standards without requiring retraining or fine-tuning. Through experimental validation and real-world case studies, we demonstrate that inference functions improve the fairness and inclusivity of LLMs while introducing minimal performance trade-offs. As AI continues to evolve, inference functions represent a key tool for promoting responsible AI and ensuring compliance with emerging ethical and regulatory frameworks.

REFERENCES

[1] **Bowman, S. R., et al. "Mitigating Bias in Large Language Models."** Proceedings of the AAAI Conference on Artificial Intelligence, 2023.

[2] **Garg, S., et al. "Adversarial Debiasing for Pre-Trained Language Models."** ArXiv preprint arXiv:2309.02705, 2023.

[3] **Zhao, J., et al. "Gender Bias in LLMs: Mitigation Techniques and Their Impact."** International Journal of AI Ethics, 2022.

[4] **Sheng, E., et al. "The Woman Worked as a Babysitter: On Biases in Language Generation."** Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2019.
[5] **Caliskan, A., et al. "Semantics Derived Automatically from Language Corpora Contain Human-like Biases."** Science, 2017.
[6] **Bender, E. M., et al. "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?"** Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 2021.
[7] **Liang, P. P., et al. "Towards Understanding and Mitigating Social Biases in Language Models."** Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020.
[8] **Mehrabi, N., et al. "A Survey on Bias and Fairness in Machine Learning."** ACM Computing Surveys, 2021.
[9] **Blodgett, S. L., et al. "Language (Technology) is Power: A Critical Survey of 'Bias' in NLP."** Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020.
[10] **Henderson, P., et al. "Ethical Challenges in Data-Driven Dialogue Systems."** Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, 2018.
[11] **Wei, J., et al. "Finetuned Language Models are Zero-Shot Learners."** ArXiv preprint arXiv:2109.01652, 2021.
[12] **Dhamala, J., et al. "Does Robustness Improve Fairness? Approaching Fairness with Robustness in NLP Models."** Findings of the Association for Computational Linguistics: EMNLP 2021, 2021.
[13] **Prates, M. O. R., et al. "Assessing Gender Bias in Machine Translation: A Case Study with Google Translate."** NeurIPS Workshop on Ethics in NLP, 2019.
[14] **Kiritchenko, S., and Mohammad, S. M. "Examining Gender and Race Bias in Two Hundred Sentiment Analysis Systems."** Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics, 2018.
[15] **Barocas, S., et al. "Fairness and Machine Learning: Limitations and Opportunities."** Book Draft, 2019.
[16] **Sun, T., et al. "Mitigating Gender Bias in Natural Language Processing: Literature Review."** Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019.
[17] **Dixon, L., et al. "Measuring and Mitigating Unintended Bias in Text Classification."** Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, 2018.
[18] **Holstein, K., et al. "Improving Fairness in Machine Learning Systems: What Do Industry Practitioners Need?"** Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 2019.
[19] **Jiang, Q., et al. "Debiasing Pre-trained Language Models."** ArXiv preprint arXiv:2007.11769, 2020.
[20] **Gonen, H., and Goldberg, Y. "Lipstick on a Pig: Debiasing Methods Cover up Systematic Gender Biases in Word Embeddings but do not Remove Them."** Proceedings of the 2019 Workshop on Fairness, Accountability, Transparency, Ethics in NLP, 2019.
[21] **Wang, T., et al. "Double-Hard Debias: Tailoring Word Embeddings for Gender Bias Mitigation."** Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019.
[22] **Zhou, D., et al. "A Closer Look at Gender Bias in Pre-trained Neural Language Models."** Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020.
[23] **Bhardwaj, R., et al. "Investigating Gender Bias in BERT."** Proceedings of the Second Workshop on Gender Bias in NLP, 2020.
[24] **Liu, L., et al. "Uncovering Gender Bias in Word-Level Language Models."** Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020.
[25] **Raji, I. D., and Buolamwini, J. "Actionable Auditing: Investigating the Impact of Publicly Naming Biased Performance Results of Commercial AI Products."** Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, 2019.
[26] **Sap, M., et al. "Social Bias Frames: Reasoning about Social and Power Implications of Language."** Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020.