The Journal of Open Source Software

LangFair: A Python Package for Assessing Bias and Fairness in Large Language Model Use Cases

Dylan Bouchard [1], **Mohit Singh Chauhan** [1], **David Skarbrevik** [1], **Viren Bajaj** [1], **and Zeya Ahmad** [1]

1 CVS Health Corporation

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Large Language Models (LLMs) have been observed to exhibit bias in numerous ways, potentially creating or worsening outcomes for specific groups identified by protected attributes such as sex, race, sexual orientation, or age. The versatile capabilities of contemporary LLMs in executing a range of tasks, as highlighted in recent studies (Liu et al., 2023; Minaee et al., 2024; Ray, 2023), pose significant challenges in assessing bias and fairness at the model level.

To help address this gap, we introduce LangFair, an open-source initiative that aims to equip LLM practitioners with the tools to evaluate bias and fairness risks relevant to their specific use cases. This evaluation method is distinctive as it incorporates actual prompts from the practitioner's use case, offering a customized assessment that accounts for prompt-specific risks that have been shown to substantially increase the probability of biased and unfair outcomes (Wang et al., 2023). While a comprehensive discussion of selection of bias and fairness metrics is outside the scope of this paper, we direct the reader to our companion paper, (Bouchard, 2024), for further details.

This paper details the accompanying Python library, `langfair`, which enables practitioners to implement the aforementioned framework in a low-code, user-friendly fashion.[1] The library offers functionality to easily generate evaluation datasets, comprised of LLM responses to use-case-specific prompts, and subsequently calculate applicable metrics for the practitioner's use case. Following (Bouchard, 2024), evaluation metrics are categorized according to the risks they assess (toxicity, stereotypes, counterfactual unfairness, and allocational harms), as well as the use case task (text generation, classification, and recommendation).[2]

Statement of Need

Traditional machine learning (ML) fairness toolkits like AIF360 (Bellamy et al., 2018), Fairlearn (Weerts et al., 2023), Aequitas (Saleiro et al., 2018) and others (Tensorflow, n.d.; Vasudevan & Kenthapadi, 2020; Wexler et al., 2019) have laid crucial groundwork but are not tailored to the generative and context-dependent nature of LLMs.

LLMs are used in systems that solve tasks such as recommendation, classification, text generation, and summarization. In practice, these systems try to restrict the responses of the LLM to the task at hand, often by including task-specific instructions in system or user prompts. When the LLM is evaluated without taking the set of task-specific prompts into account, the evaluation metrics are not representative of the system's true performance. Representing the system's actual performance is especially important when evaluating its outputs for bias and

[1] The repository for `langfair` can be found at https://github.com/cvs-health/langfair.

[2] Note that text generation encompasses all use cases for which output is text, but does not belong to a predefined set of elements (as with classification and recommendation).

fairness risks because they pose real harm to the user and, by way of repercussions, the system developer.

Most evaluation tools, including those that assess bias and fairness risk, evaluate LLMs at the model-level by calculating metrics based on the responses of the LLMs to static benchmark datasets of prompts (Bartl et al., 2020; Dev et al., 2019; Dhamala et al., 2021; Gehman et al., 2020; HowieHwong, n.d.; Katyfelkner, n.d.; Krieg et al., 2022; Levy et al., 2021; Li et al., 2020; Nadeem et al., 2020; Nangia et al., 2020; Nozza et al., 2021; Nyu-Mll, n.d.; Rudinger et al., 2018; *Saif | Bias EEC*, n.d.; Smith et al., 2022; Umanlp, n.d.; Vnmssnhv, n.d.; Webster et al., 2018; Zhao et al., 2018) that do not consider prompt-specific risks and are often independent of the task at hand. Holistic Evaluation of Language Models (HELM) (Liang et al., 2023), DecodingTrust (Wang et al., 2023), and several others (authors, 2023; Gao et al., 2024; Y. Huang et al., 2024) follow this paradigm. Some tools allow you to configure the evaluation to specific but predefined tasks such as LightEval (Fourrier et al., 2023), langtest (Nazir et al., 2024), and others (Confident-Ai, n.d.; Giskard-Ai, n.d.; Huggingface, n.d.).

LangFair complements the aforementioned frameworks because it follows a bring your own prompts (BYOP) approach, which allows users to tailor the bias and fairness evaluation to their use case by computing metrics using LLM responses to user-provided prompts. This addresses the need for a task-based bias and fairness evaluation tool that accounts for prompt-specific risk for LLMs.[3]

LangFair addresses another challenge faced by developers: navigating the large number of bias and fairness metrics to find the ones that apply to their use case. While the aforementioned collection of existing tools offer extensive metric coverage, LangFair offers an actionable decision framework to guide metric selection (Bouchard, 2024). The package is designed based on this decision framework to derive metrics that are applicable to recommendation, classification, and text generation tasks based on use-case-specific properties such as fairness through unawareness (Gallegos et al., 2024), inputs corresponding to protected attribute groups, etc.

Furthermore, LangFair is designed for real-world LLM-based systems that require governance audits. LangFair focuses on calculating metrics from LLM responses only, which is more practical for real-world testing where access to internal states of model to retrieve embeddings or token probabilities is difficult. An added benefit is that output-based metrics, which are focused on the downstream task, have shown to be potentially more reliable than metrics derived from embeddings or token probabilities [Goldfarb-Tarrant et al. (2021); delobelle-etal-2022-measuring].

Generation of Evaluation Datasets

Following (Bouchard, 2024), we define bias and fairness assessments for LLMs on a use case level. Under this approach, evaluation metrics are computed on a set of LLM responses generated from prompts sampled from the use case's population of prompts. Accordingly, the `langfair.generator` module offers two classes, `ResponseGenerator` and `CounterfactualGenerator`, which aim to enable user-friendly construction of evaluation datasets for text generation use cases.

`ResponseGenerator` class

To streamline generation of evaluation datasets, the `ResponseGenerator` class wraps an instance of a `langchain` LLM and leverages asynchronous generation with `asyncio`. Users should customize the `langchain` LLM instance to match the parameters of the use case being assessed. To implement, users simply pass a list of prompts

[3] Experiments in (Wang et al., 2023) demonstrate that prompt content has substantial influence on the likelihood of biased LLM responses.

(strings) to the `ResponseGenerator.generate_responses` method, which returns a dictionary containing prompts, responses, and applicable metadata. In addition, the `ResponseGenerator.estimate_token_cost` method enables users to estimate the approximate cost of generation with select `OpenAI` models in advance by counting tokens with the `tiktoken` library. In particular, `tiktoken` and model-cost mapping are used to a) compute deterministic input token costs from a provided list of prompts and b) estimate stochastic output token costs from a sample of generated responses.[4]

`CounterfactualGenerator` class

Counterfactual fairness assessments are recommended Bouchard (2024) for text generation use cases that do not satisfy fairness through unawareness (FTU), i.e., prompts contain mentions of protected attribute information (Gallegos et al., 2024). In the context of LLMs, counterfactual fairness can be assessed by constructing counterfactual input pairs Bouchard (2024), comprised of prompt pairs that mention different protected attribute groups but are otherwise identical, and measuring the differences in the corresponding generated output pairs.

A subclass of `ResponseGenerator`, the `CounterfactualGenerator` offers functionality to check for FTU, construct counterfactual input pairs, and generate corresponding pairs of responses asynchronously using a `langchain` LLM instance. Off the shelf, the FTU check and creation of counterfactual input pairs can be done for gender and race/ethnicity, but users may also provide a custom mapping of protected attribute words to enable this functionality for other attributes as well.[5]

Bias and Fairness Evaluations for Focused Use Cases

The evaluation metrics supported by LangFair assess the following bias and fairness risks: toxicity, stereotypes, counterfactual (un)fairness, and allocational harms. Table 1 maps the classes contained in the `langfair.metrics` module to these risks. These classes are discussed in detail below.

Class	Risk Assessed	Applicable Tasks
`ToxicityMetrics`	Toxicity	Text generation
`StereotypeMetrics`	Stereotypes	Text generation
`CounterfactualMetrics`	Counterfactual fairness	Text generation
`RecommendationMetrics`	Counterfactual fairness	Recommendation
`ClassificationMetrics`	Allocational harms	Classification

Table 1 : Classes for Computing Evaluation Metrics in langfair.metrics

Toxicity Metrics

The `ToxicityMetrics` class facilitates simple computation of toxicity metrics from a user-provided list of LLM responses. These metrics include *expected maximum toxicity* (Gehman et al., 2020), *toxicity probability* (Gehman et al., 2020), and *toxic fraction* (Liang et al., 2023), all of which leverage a pre-trained toxicity classifier that maps a text input to a toxicity score ranging from 0 to 1. For off-the-shelf toxicity classifiers, the `ToxicityMetrics` class provides four options: two classifiers from the `detoxify` package, `roberta-hate-speech-dynabench-r4-target` from the `evaluate` package, and `toxigen` available on

[4] Token costs for select OpenAI models are obtained from https://openai.com/api/pricing/.

[5] For instance, one example of a custom counterfactual mapping could be `{'old': ['old', 'elderly', 'senior'], 'young': ['young', 'youthful', 'juvenile']}` To construct the counterfactual input pairs, token-based substitution is conducted on user-provided prompts. For instance, the input prompt `the husband went to the store`, would yield the counterfactual input pair `['the husband went to the store','the wife went to the store']` for gender.

HuggingFace.[https://github.com/unitaryai/detoxify][https://github.com/huggingface/evaluate][6] For additional flexibility, users can specify an ensemble of the off-the-shelf classifiers offered or provide a custom toxicity classifier object.

Stereotype Metrics

LLMs have been observed to include harmful stereotypes in their generated responses Zekun et al. (2023). To measure stereotypes in LLM responses, the `StereotypeMetrics` class offers two classes of metrics: metrics based on word cooccurrences and metrics that leverage a pre-trained stereotype classifier. In particular, metrics based on word cooccurrences include *cooccurrence bias score* (Bordia & Bowman, 2019) and *stereotypical associations* (Liang et al., 2023) and aim to assess relative cooccurrence of stereotypical words with certain protected attribute words. Stereotype classifier metrics leverage the `wu981526092/Sentence-Level-Stereotype-Detector` classifier available on HuggingFace (Zekun et al., 2023) and compute analogs of the aforementioned toxicity classifier metrics (Bouchard, 2024).^[https://huggingface.co/wu981526092/Sentence-Level-Stereotype-Detector

Counterfactual Fairness Metrics for Text Generation

The `CounterfactualMetrics` class offers two groups of metrics to assess counterfactual fairness in text generation use cases. The first set of metrics leverage a pre-trained sentiment classifier to measure sentiment disparities in counterfactually generated outputs (Bouchard, 2024; P.-S. Huang et al., 2020). This class uses the `vaderSentiment` classifier by default but also gives users the option to provide a custom sentiment classifier object.[7] The second group of metrics addresses a stricter desiderata and measures overall similarity in counterfactually generated outputs. Following (Bouchard, 2024), these metrics apply well-established text similarity metrics including *recall-oriented understudy for gisting evaluation (ROUGE)* (Lin, 2004), *bilingual evaluation understudy (BLEU)* (Papineni et al., 2002), and *cosine similarity* to measure counterfactual similarity.

Counterfactual Fairness Metrics for Recommendation

When LLMs are used for recommendation, they pose the risk of discriminating when exposed to protected attribute information in input prompts (J. Zhang et al., 2023). To assess counterfactual fairness for recommendation use cases, the `RecommendationMetrics` class offers three metrics, proposed by (J. Zhang et al., 2023). Specifically, for counterfactually generated sets of K recommendations, these metrics include *Jaccard-K*, *pairwise ranking accuracy gap* (*PRAG-K*), and *search result page misinformation score* (*SERP-K*). Metrics may be computed pairwise (Bouchard, 2024), or attribute-wise (J. Zhang et al., 2023).

Fairness Metrics for Classification

Allocational harms, as measured by group fairness metrics for classification models, has been widely studied in the machine learning fairness literature B. H. Zhang et al. (2018). When LLMs are used to solve classification problems, traditional machine learning fairness metrics may be applied, provided that inputs can be mapped to a protected attribute. To this end, the `ClassificationMetrics` class offers a suite of metrics to address unfair classification. Following the framework proposed by (Saleiro et al., 2018), metrics are segmented into three categories: representation fairness, error-based fairness for assistive use cases, and error-based fairness for punitive use cases. Representation fairness includes a single metric that measures disparity in predicted prevalence rates Saleiro et al. (2018). For assistive (punitive) classification use cases, metrics measure disparities in false negative rate and false omission rate (false positive

[6] https://github.com/microsoft/TOXIGEN
[7] https://github.com/cjhutto/vaderSentiment

rate and false discovery rate).[8] When computing metrics using the `ClassificationMetrics` class, the user may specify whether to compute these metrics as pairwise differences (Bellamy et al., 2018) or pairwise ratios (Saleiro et al., 2018).

Semi-Automated Evaluation

`AutoEval` class

To streamline assessments for text generation use cases, the `AutoEval` class conducts a multi-step process that includes metric selection, evaluation dataset generation, and metric computation. The user is required to supply a list of prompts and an instance of `langchain` LLM. Below we provide a basic example demonstrating the execution of `AutoEval.evaluate` with a `gemini-pro` instance.[9]

```
from langchain_google_vertexai import VertexAI
from langfair.auto import AutoEval

llm = VertexAI(model_name='gemini-pro')
auto_object = AutoEval(prompts=prompts, langchain_llm=llm)
results = await auto_object.evaluate()
```

Under the hood, the `AutoEval.evaluate` method 1) checks for FTU, 2) generates responses and counterfactual responses (if FTU is not satisfied), and 3) calculates applicable metrics for the use case.[10] This process flow is depicted in Figure 1.

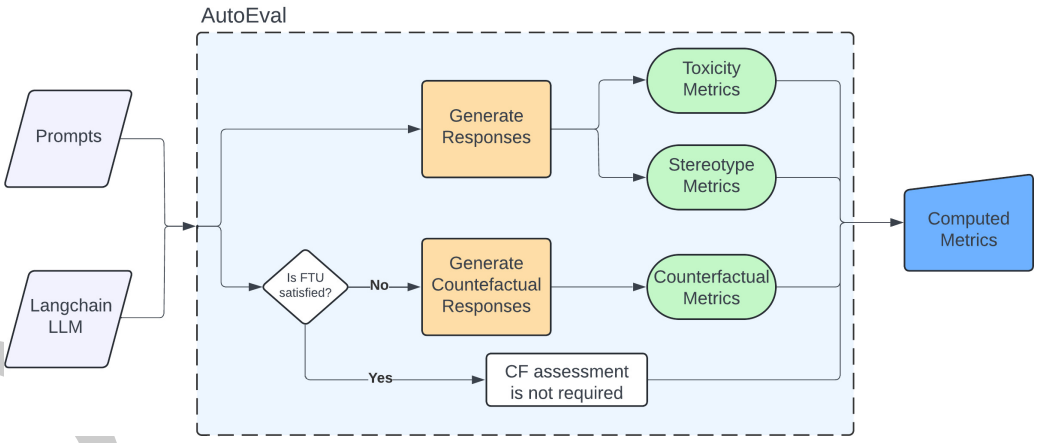

Figure 1:Flowchart of internal design of Autoeval.evaluate method

Author Contributions

Dylan Bouchard was the principal developer and researcher of the LangFair project, responsible for conceptualization, methodology, and software development of the *langfair* library. Mohit Singh Chauhan was the architect behind the structural design of the *langfair* library and helped lead the software development efforts. David Skarbrevik was the primary author of LangFair's documentation, helped implement software engineering best practices, and contributed to

[8] In the context of fairness, false negatives are especially costly for use cases that are assistive in nature, e.g. qualifying for a benefits program, . Hence, the recommended metrics for these use cases assess disparities in false negatives across groups. Conversely, false positives are highly costly in punitive use cases (e.g. fraud prediction), and hence the associated metrics focus on disparities in false positives across groups.

[9] Note that this example assumes the user has already set up their VertexAI credentials and sampled a list of prompts from their use case prompts.

[10] The 'AutoEval' class is designed specifically for text generation use cases. Applicable metrics include toxicity metrics, stereotype metrics, and, if FTU is not satisfied, counterfactual fairness metrics.

software development. Viren Bajaj wrote unit tests, contributed to the software development, and helped implement software engineering best practices. Zeya Ahmad contributed to the software development.

Acknowledgements

We wish to thank Piero Ferrante, Blake Aber, Xue (Crystal) Gu, and Zirui Xu for their helpful suggestions.

References

Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., & Wallach, H. (2018). *A reductions approach to fair classification*. https://arxiv.org/abs/1803.02453

authors, B. (2023). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*. https://openreview.net/forum?id=uyTL5Bvosj

Bartl, M., Nissim, M., & Gatt, A. (2020). Unmasking contextual stereotypes: Measuring and mitigating BERT's gender bias. In M. R. Costa-jussà, C. Hardmeier, K. Webster, & W. Radford (Eds.), *Proceedings of the second workshop on gender bias in natural language processing*.

Bellamy, R. K. E., Dey, K., Hind, M., Hoffman, S. C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilovic, A., Nagar, S., Ramamurthy, K. N., Richards, J., Saha, D., Sattigeri, P., Singh, M., Varshney, K. R., & Zhang, Y. (2018). *AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias*. https://arxiv.org/abs/1810.01943

Bordia, S., & Bowman, S. R. (2019). Identifying and reducing gender bias in word-level language models. In S. Kar, F. Nadeem, L. Burdick, G. Durrett, & N.-R. Han (Eds.), *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Student research workshop* (pp. 7–15). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-3002

Bouchard, D. (2024). *An actionable framework for assessing bias and fairness in large language model use cases*. https://arxiv.org/abs/2407.10853

Confident-Ai. (n.d.). *GitHub - confident-ai/deepeval: The LLM Evaluation Framework*. https://github.com/confident-ai/deepeval

Dev, S., Li, T., Phillips, J., & Srikumar, V. (2019). *On measuring and mitigating biased inferences of word embeddings*. https://arxiv.org/abs/1908.09369

Dhamala, J., Sun, T., Kumar, V., Krishna, S., Pruksachatkun, Y., Chang, K.-W., & Gupta, R. (2021). BOLD: Dataset and metrics for measuring biases in open-ended language generation. *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 862–872. https://doi.org/10.1145/3442188.3445924

Feldman, M., Friedler, S., Moeller, J., Scheidegger, C., & Venkatasubramanian, S. (2015). *Certifying and removing disparate impact*. https://arxiv.org/abs/1412.3756

Fourrier, C., Habib, N., Wolf, T., & Tunstall, L. (2023). *LightEval: A lightweight framework for LLM evaluation* (Version 0.3.0). https://github.com/huggingface/lighteval

Gallegos, I. O., Rossi, R. A., Barrow, J., Tanjim, M. M., Kim, S., Dernoncourt, F., Yu, T., Zhang, R., & Ahmed, N. K. (2024). *Bias and fairness in large language models: A survey*. https://arxiv.org/abs/2309.00770

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., … Zou, A. (2024). *A framework for few-shot language model evaluation* (Version v0.4.3). Zenodo. https://doi.org/10.5281/zenodo.12608602

Gehman, S., Gururangan, S., Sap, M., Choi, Y., & Smith, N. A. (2020). RealToxicityPrompts: Evaluating neural toxic degeneration in language models. *Findings*. https://api.semanticscholar.org/CorpusID:221878771

Giskard-Ai. (n.d.). *GitHub - giskard-AI/giskard: Open-source evaluation & testing for ML models & LLMs*. https://github.com/Giskard-AI/giskard

Goldfarb-Tarrant, S., Marchant, R., Sanchez, R. M., Pandya, M., & Lopez, A. (2021). *Intrinsic bias metrics do not correlate with application bias*. https://arxiv.org/abs/2012.15859

Hardt, M., Price, E., & Srebro, N. (2016). *Equality of opportunity in supervised learning*. https://arxiv.org/abs/1610.02413

HowieHwong. (n.d.). *GitHub - HowieHwong/TrustGPT: Can we Trust Large Language Models?: A benchmark for responsible large language models via toxicity, Bias, and Value-alignment evaluation*. https://github.com/HowieHwong/TrustGPT

Huang, P.-S., Zhang, H., Jiang, R., Stanforth, R., Welbl, J., Rae, J., Maini, V., Yogatama, D., & Kohli, P. (2020). *Reducing sentiment bias in language models via counterfactual evaluation*. https://arxiv.org/abs/1911.03064

Huang, Y., Sun, L., Wang, H., Wu, S., Zhang, Q., Li, Y., Gao, C., Huang, Y., Lyu, W., Zhang, Y., Li, X., Sun, H., Liu, Z., Liu, Y., Wang, Y., Zhang, Z., Vidgen, B., Kailkhura, B., Xiong, C., … Zhao, Y. (2024). TrustLLM: Trustworthiness in large language models. *Forty-First International Conference on Machine Learning*. https://openreview.net/forum?id=bWUU0LwwMp

Huggingface. (n.d.). *GitHub - huggingface/evaluate: Evaluate: A library for easily evaluating machine learning models and datasets*. https://github.com/huggingface/evaluate

Kamiran, F., Karim, A., & Zhang, X. (2012). Decision theory for discrimination-aware classification. *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, 924–929. https://doi.org/10.1109/ICDM.2012.45

Kamishima, T., Akaho, S., Asoh, H., & Sakuma, J. (2012). Fairness-aware classifier with prejudice remover regularizer. *Proceedings of the 2012th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II*, 35–50. ISBN: 9783642334856

Katyfelkner. (n.d.). *GitHub - katyfelkner/winoqueer*. https://github.com/katyfelkner/winoqueer

Krieg, K., Parada-Cabaleiro, E., Medicus, G., Lesota, O., Schedl, M., & Rekabsaz, N. (2022). Grep-BiasIR: A dataset for investigating gender representation-bias in information retrieval results. *Proceeding of the 2023 ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR)*.

Levy, S., Lazar, K., & Stanovsky, G. (2021). *Collecting a large-scale gender bias dataset for coreference resolution and machine translation*. https://arxiv.org/abs/2109.03858

Li, T., Khot, T., Khashabi, D., Sabharwal, A., & Srikumar, V. (2020). UnQovering stereotyping biases via underspecified questions. *Findings of EMNLP*.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., Newman, B., Yuan, B., Yan, B., Zhang, C., Cosgrove, C., Manning, C. D., Ré, C., Acosta-Navas, D., Hudson, D. A., … Koreeda, Y. (2023). *Holistic evaluation of language models*. https://arxiv.org/abs/2211.09110

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 74–81. https://aclanthology.org/W04-1013

Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., Wu, Z., Zhao, L., Zhu, D., Li, X., Qiang, N., Shen, D., Liu, T., & Ge, B. (2023). Summary of ChatGPT-related research and perspective towards the future of large language models. *Meta-Radiology*, *1*(2), 100017. https://doi.org/10.1016/j.metrad.2023.100017

Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., & Gao, J. (2024). *Large language models: A survey*. https://arxiv.org/abs/2402.06196

Nadeem, M., Bethke, A., & Reddy, S. (2020). *StereoSet: Measuring stereotypical bias in pretrained language models*. https://arxiv.org/abs/2004.09456

Nangia, N., Vania, C., Bhalerao, R., & Bowman, S. R. (2020, November). CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Nazir, A., Chakravarthy, T. K., Cecchini, D. A., Chakravarthy, T. K., Khajuria, R., Sharma, P., Mirik, A. T., Kocaman, V., & Talby, D. (2024). LangTest: A comprehensive evaluation library for custom LLM and NLP models. *Software Impacts*, *19*(100619). https://doi.org/10.1016/j.simpa.2024.100619

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

Nyu-Mll. (n.d.). *GitHub - nyu-mll/BBQ: Repository for the Bias Benchmark for QA dataset*. https://github.com/nyu-mll/BBQ

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 311–318. https://doi.org/10.3115/1073083.1073135

Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., & Weinberger, K. Q. (2017). *On fairness and calibration*. https://arxiv.org/abs/1709.02012

Ray, P. P. (2023). ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, *3*, 121–154. https://doi.org/https://doi.org/10.1016/j.iotcps.2023.04.003

Rudinger, R., Naradowsky, J., Leonard, B., & Van Durme, B. (2018). Gender bias in coreference resolution. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Saif | Bias EEC. (n.d.). http://saifmohammad.com/WebPages/Biases-SA.html

Saleiro, P., Kuester, B., Stevens, A., Anisfeld, A., Hinkson, L., London, J., & Ghani, R. (2018). Aequitas: A bias and fairness audit toolkit. *arXiv Preprint arXiv:1811.05577*.

Smith, E. M., Hall, M., Kambadur, M., Presani, E., & Williams, A. (2022). *"I'm sorry to hear that": Finding new biases in language models with a holistic descriptor dataset*. https://doi.org/10.48550/ARXIV.2205.09209

Tensorflow. (n.d.). *GitHub - tensorflow/fairness-indicators: Tensorflow's Fairness Evaluation and Visualization Toolkit*. https://github.com/tensorflow/fairness-indicators

Umanlp. (n.d.). *GitHub - umanlp/RedditBias: Code & Data for the paper "RedditBias: A Real-World Resource for Bias Evaluation and Debiasing of Conversational Language Models"*. https://github.com/umanlp/RedditBias

Vasudevan, S., & Kenthapadi, K. (2020). *The LinkedIn fairness toolkit (LiFT)*. https://github.

com/linkedin/lift.

Vnmssnhv. (n.d.). *GitHub - vnmssnhv/NeuTralRewriter: Neutral rewriter*. https://github.com/vnmssnhv/NeuTralRewriter

Wang, B., Chen, W., Pei, H., Xie, C., Kang, M., Zhang, C., Xu, C., Xiong, Z., Dutta, R., Schaeffer, R., & others. (2023). *DecodingTrust: A comprehensive assessment of trustworthiness in GPT models*.

Webster, K., Recasens, M., Axelrod, V., & Baldridge, J. (2018). Mind the GAP: A balanced corpus of gendered ambiguou. *Transactions of the ACL*, to appear.

Weerts, H., Dudík, M., Edgar, R., Jalali, A., Lutz, R., & Madaio, M. (2023). Fairlearn: Assessing and Improving Fairness of AI Systems. *Journal of Machine Learning Research*, *24*. http://jmlr.org/papers/v24/23-0389.html

Wexler, J., Pushkarna, M., Bolukbasi, T., Wattenberg, M., Viégas, F. B., & Wilson, J. (2019). The what-if tool: Interactive probing of machine learning models. *CoRR*, *abs/1907.04135*. http://arxiv.org/abs/1907.04135

Zekun, W., Bulathwela, S., & Koshiyama, A. S. (2023). *Towards auditing large language models: Improving text-based stereotype detection*. https://arxiv.org/abs/2311.14126

Zhang, B. H., Lemoine, B., & Mitchell, M. (2018). *Mitigating unwanted biases with adversarial learning*. https://arxiv.org/abs/1801.07593

Zhang, J., Bao, K., Zhang, Y., Wang, W., Feng, F., & He, X. (2023). Is ChatGPT fair for recommendation? Evaluating fairness in large language model recommendation. *Proceedings of the 17th ACM Conference on Recommender Systems*, *2012*, 993–999. https://doi.org/10.1145/3604915.3608860

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., & Chang, K.-W. (2018). *Gender Bias in Coreference Resolution: Evaluation and Debiasing methods*. https://arxiv.org/abs/1804.06876