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: 10.21105/joss.07570

Software

- Review
- Repository
- Archive

Editor: Chris Vernon

Reviewers:

- @xavieryao
- @emily-sexton

Submitted: 03 December 2024
Published: 23 January 2025

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. To help address this gap, we introduce `langfair`, an open-source Python package that aims to equip LLM practitioners with the tools to evaluate bias and fairness risks relevant to their specific use cases.[1] The package 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. To guide in metric selection, LangFair offers an actionable decision framework, discussed in detail in the project's companion paper, Bouchard (2024).

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, 2020; Vasudevan & Kenthapadi, 2020; Wexler et al., 2019) have laid crucial groundwork. These toolkits offer various metrics and algorithms that focus on assessing and mitigating bias and fairness through different stages of the ML lifecycle. While the fairness assessments offered by these toolkits include a wide variety of generic fairness metrics, which can also apply to certain LLM use cases, they are not tailored to the generative and context-dependent nature of LLMs.[2]

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 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 (Barikeri et al., 2021; Bartl et al., 2020; Dhamala et al., 2021; Felkner et al., 2024; Gehman et al., 2020; Y. Huang et al., 2023; Kiritchenko & Mohammad, 2018; Krieg et al., 2023; Levy et al., 2021; Li et al., 2020; Nadeem et al., 2020; Nangia et al., 2020; Nozza et al., 2021; Parrish et al., 2022; Qian et al., 2022; Rudinger et al., 2018; Webster et al.,

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

[2] The toolkits mentioned here offer fairness metrics for classification. In a similar vein, the recommendation fairness metrics offered in FaiRLLM (Zhang et al., 2023) can be applied to ML recommendation systems as well as LLM recommendation use cases.

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 other toolkits (Gao et al., 2024; Y. Huang et al., 2024; Huggingface, 2022; Nazir et al., 2024; Srivastava et al., 2022) follow this paradigm.

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]

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 (Delobelle et al., 2022; Goldfarb-Tarrant et al., 2021).

Generation of Evaluation Datasets

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`. To implement, users simply pass a list of prompts (strings) to the `ResponseGenerator.generate_responses` method, which returns a dictionary containing prompts, responses, and applicable metadata.

`CounterfactualGenerator` class

In the context of LLMs, counterfactual fairness can be assessed by constructing counterfactual input pairs (Bouchard, 2024; Gallegos et al., 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. These assessments are applicable to use cases that do not satisfy fairness through unawareness (FTU), meaning prompts contain mentions of protected attribute groups. To address this, the `CounterfactualGenerator` class offers functionality to check for FTU, construct counterfactual input pairs, and generate corresponding pairs of responses asynchronously using a `langchain` LLM instance.[4] 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.

Bias and Fairness Evaluations for Focused Use Cases

Following Bouchard (2024), evaluation metrics are categorized according to the risks they assess (toxicity, stereotypes, counterfactual unfairness, and allocational harms), as well as

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

[4]In practice, a FTU check consists of parsing use case prompts for mentions of protected attribute groups.

The Journal of Open Source Software

the use case task (text generation, classification, and recommendation).[5] 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 leverage a pre-trained toxicity classifier that maps a text input to a toxicity score ranging from 0 to 1 (Gehman et al., 2020; Liang et al., 2023). 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 HuggingFace.[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

To measure stereotypes in LLM responses, the `StereotypeMetrics` class offers two categories of metrics: metrics based on word cooccurrences and metrics that leverage a pre-trained stereotype classifier. Metrics based on word cooccurrences aim to assess relative cooccurrence of stereotypical words with certain protected attribute words. On the other hand, stereotype-classifier-based metrics leverage the `wu981526092/Sentence-Level-Stereotype-Detector` classifier available on HuggingFace (Zekun et al., 2023) and compute analogs of the aforementioned toxicity-classifier-based metrics (Bouchard, 2024).[7]

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 group of metrics leverage a pre-trained sentiment classifier to measure sentiment disparities in counterfactually generated outputs (see P.-S. Huang et al. (2020) for further details). This class uses the `vaderSentiment` classifier by default but also gives users the option to provide a custom sentiment classifier object.[8] The second group of metrics addresses a stricter desiderata and measures overall similarity in counterfactually generated outputs using well-established text similarity metrics (Bouchard, 2024).

Counterfactual Fairness Metrics for Recommendation

The `RecommendationMetrics` class is designed to assess counterfactual fairness for recommendation use cases. Specifically, these metrics measure similarity in generated lists of recommendations from counterfactual input pairs. Metrics may be computed pairwise (Bouchard, 2024), or attribute-wise (Zhang et al., 2023).

[5]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).
[6]https://github.com/unitaryai/detoxify; https://github.com/huggingface/evaluate; https://github.com/microsoft/TOXIGEN
[7]https://huggingface.co/wu981526092/Sentence-Level-Stereotype-Detector
[8]https://github.com/cjhutto/vaderSentiment

The Journal of Open Source Software

Fairness Metrics for Classification

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 by measuring disparities in predicted prevalence, false negatives, or false positives. 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 (each step is described in detail above) for a comprehensive fairness assessment. Specifically, these steps include metric selection (based on whether FTU is satsified), evaluation dataset generation from user-provided prompts with a user-provided LLM, and computation of applicable fairness metrics. To implement, 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 ChatVertexAI
from langfair.auto import AutoEval

llm = ChatVertexAI(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.

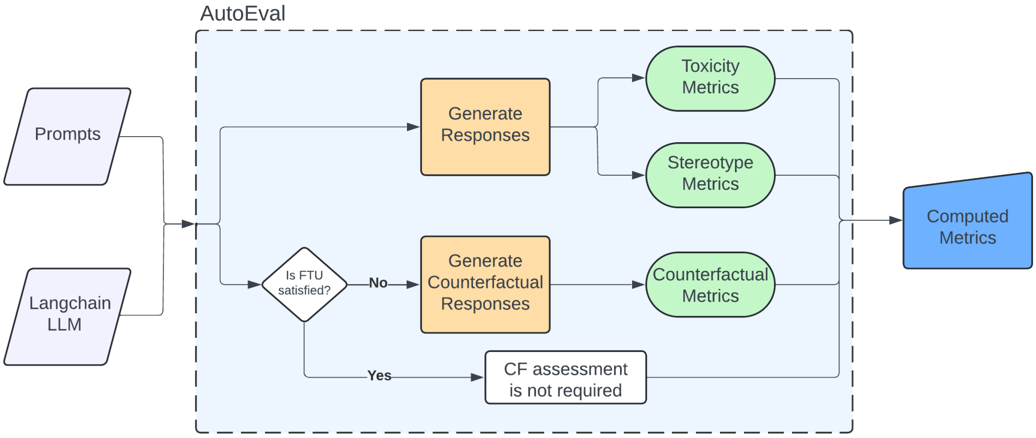

Figure 1:Flowchart of internal design of Autoeval.evaluate method

[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.

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 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

Barikeri, S., Lauscher, A., Vulić, I., & Glavaš, G. (2021). *RedditBias: A real-world resource for bias evaluation and debiasing of conversational language models*. https://doi.org/10.48550/arXiv.2106.03521

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*. https://doi.org/10.48550/arXiv.2010.14534

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://doi.org/10.48550/arXiv.1810.01943

Bouchard, D. (2024). *An actionable framework for assessing bias and fairness in large language model use cases*. https://doi.org/10.48550/arXiv.2407.10853

Delobelle, P., Tokpo, E., Calders, T., & Berendt, B. (2022). Measuring fairness with biased rulers: A comparative study on bias metrics for pre-trained language models. In M. Carpuat, M.-C. de Marneffe, & I. V. Meza Ruiz (Eds.), *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: Human language technologies* (pp. 1693–1706). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.122

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

Felkner, V. K., Chang, H.-C. H., Jang, E., & May, J. (2024). *WinoQueer: A community-in-the-loop benchmark for anti-LGBTQ+ bias in large language models*. https://doi.org/10.48550/arXiv.2306.15087

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://doi.org/10.48550/arXiv.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://doi.org/10.18653/v1/2020.findings-emnlp.301

Goldfarb-Tarrant, S., Marchant, R., Sanchez, R. M., Pandya, M., & Lopez, A. (2021). *Intrinsic bias metrics do not correlate with application bias*. https://doi.org/10.18653/v1/2021.acl-long.150

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://doi.org/10.18653/v1/2020.findings-emnlp.7

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://doi.org/10.48550/arXiv.2401.05561

Huang, Y., Zhang, Q., Y, P. S., & Sun, L. (2023). *TrustGPT: A benchmark for trustworthy and responsible large language models*. https://doi.org/10.48550/arXiv.2306.11507

Huggingface. (2022). *GitHub - huggingface/evaluate: Evaluate: A library for easily evaluating machine learning models and datasets*. https://github.com/huggingface/evaluate

Kiritchenko, S., & Mohammad, S. M. (2018). *Examining gender and race bias in two hundred sentiment analysis systems*. https://doi.org/10.18653/v1/S18-2005

Krieg, K., Parada-Cabaleiro, E., Medicus, G., Lesota, O., Schedl, M., & Rekabsaz, N. (2023). Grep-BiasIR: A dataset for investigating gender representation bias in information retrieval results. *Proceedings of the 2023 Conference on Human Information Interaction and Retrieval*, 444–448. https://doi.org/10.1145/3576840.3578295

Levy, S., Lazar, K., & Stanovsky, G. (2021). *Collecting a large-scale gender bias dataset for coreference resolution and machine translation*. https://doi.org/10.48550/arXiv.2109.03858

Li, T., Khashabi, D., Khot, T., Sabharwal, A., & Srikumar, V. (2020). UNQOVERing stereotyping biases via underspecified questions. In T. Cohn, Y. He, & Y. Liu (Eds.), *Findings of the association for computational linguistics: EMNLP 2020* (pp. 3475–3489). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.311

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://doi.org/10.48550/arXiv.2211.09110

Nadeem, M., Bethke, A., & Reddy, S. (2020). *StereoSet: Measuring stereotypical bias in pretrained language models*. https://doi.org/10.48550/arXiv.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*. https://doi.org/10.48550/arXiv.2010.00133

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

Parrish, A., Chen, A., Nangia, N., Padmakumar, V., Phang, J., Thompson, J., Htut, P. M., & Bowman, S. (2022). BBQ: A hand-built bias benchmark for question answering. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Findings of the association for computational linguistics: ACL 2022* (pp. 2086–2105). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-acl.165

Qian, R., Ross, C., Fernandes, J., Smith, E., Kiela, D., & Williams, A. (2022). *Perturbation augmentation for fairer NLP*. https://doi.org/10.48550/arXiv.2205.12586

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*. https://doi.org/10.48550/arXiv.1804.09301

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*. https://doi.org/10.48550/arXiv.1811.05577

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., & others. (2022). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv Preprint arXiv:2206.04615*. https://doi.org/10.48550/arXiv.2206.04615

Tensorflow. (2020). *GitHub - tensorflow/fairness-indicators: Tensorflow's Fairness Evaluation and Visualization Toolkit*. https://github.com/tensorflow/fairness-indicators

Vasudevan, S., & Kenthapadi, K. (2020). LiFT: A scalable framework for measuring fairness in ML applications. *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. https://doi.org/10.1145/3340531.3412705

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*. https://doi.org/10.48550/arXiv.2306.11698

Webster, K., Recasens, M., Axelrod, V., & Baldridge, J. (2018). Mind the GAP: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, *6*, 605–617. https://doi.org/10.1162/tacl_a_00240

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*. https://doi.org/10.1109/TVCG.2019.2934619

Zekun, W., Bulathwela, S., & Koshiyama, A. S. (2023). *Towards auditing large language models: Improving text-based stereotype detection*. https://doi.org/10.48550/arXiv.2311.14126

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://doi.org/10.48550/arXiv.1804.06876