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*

1 June 2024

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 [34, 33, 42], 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 [50]. 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, [8], 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 [8], 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]

[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).

Statement of Need

Traditional machine learning (ML) fairness toolkits like AIF360 [4], Fairlearn [52], Aequitas [44] and others [48, 53, 46, 6] 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 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 [43, 57, 49, 51, 29, 35, 3, 36, 27, 47, 1, 11, 17, 12, 45, 21, 38, 39, 30, 28] that do not consider prompt-specific risks and are often independent of the task at hand. Holistic Evaluation of Language Models (HELM) [31], DecodingTrust [50], and several others [5, 23, 16] follow this paradigm. Some tools allow you to configure the evaluation to specific but predefined tasks such as LightEval [14], langtest [37], and others [9, 18, 24].

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 [8]. 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 [15], 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,

[3] Experiments in [50] demonstrate that prompt content has substantial influence on the likelihood of biased LLM responses.

have shown to be potentially more reliable than metrics derived from embeddings or token probabilities [19, 10].

Generation of Evaluation Datasets

Following [8], 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 (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 [22, 8] for text generation use cases that do not satisfy fairness through unawareness (FTU), i.e., prompts contain mentions of protected attribute information [15]. In the context of LLMs, counterfactual fairness can be assessed by constructing counterfactual input pairs [15, 8], 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] To construct the counterfactual input pairs, token-based sub-

[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']}`.

stitution 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.

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.

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

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

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* [17], *toxicity probability* [17], and *toxic fraction* [31], 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 HuggingFace.[6][7][8] 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 [31, 7, 54]. 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* [7] and *stereotypical associations* [31] and aim to assess relative cooccurrence of stereotypical words with certain protected attribute words. Stereotype classifier metrics leverage the `wu981526092/Sentence-Level-Stereotype-Detector`

[6] https://github.com/unitaryai/detoxify
[7] https://github.com/huggingface/evaluate
[8] https://github.com/microsoft/TOXIGEN

classifier available on HuggingFace [54] and compute analogs of the aforementioned toxicity classifier metrics [8].[9]

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 [22, 8]. This class uses the `vaderSentiment` classifier by default but also gives users the option to provide a custom sentiment classifier object.[10] The second group of metrics addresses a stricter desiderata and measures overall similarity in counterfactually generated outputs. Following [8], these metrics apply well-established text similarity metrics including *recall-oriented understudy for gisting evaluation* (*ROUGE*) [32], *bilingual evaluation understudy* (*BLEU*) [40], 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 [56]. To assess counterfactual fairness for recommendation use cases, the `RecommendationMetrics` class offers three metrics, proposed by [56]. 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 [8], or attribute-wise [56].

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 [44, 4, 52, 13, 26, 20, 41, 25, 2, 55]. 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 [44], 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 [13, 44]. For assistive (punitive) classification use cases, metrics measure disparities in false negative rate and false omission rate (false positive rate and false discovery rate).[11] When computing metrics using

[9] https://huggingface.co/wu981526092/Sentence-Level-Stereotype-Detector

[10] https://github.com/cjhutto/vaderSentiment

[11] 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.

the `ClassificationMetrics` class, the user may specify whether to compute these metrics as pairwise differences [4] or pairwise ratios [44].

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.[12]

```
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()
```

Listing 1: `AutoEval` Example

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.[13] This process flow is depicted in Figure 1.

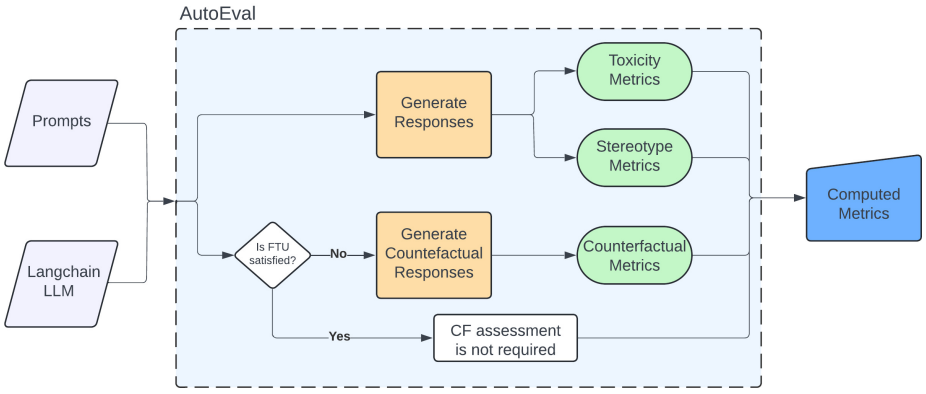

Figure 1: Flowchart of internal design of `AutoEval.evaluate` method.

[12] 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.

[13] 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

Saif | Bias EEC.

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification, 2018.

Marion Bartl, Malvina Nissim, and Albert Gatt. Unmasking contextual stereotypes: Measuring and mitigating bert's gender bias. In Marta R. Costa-jussà, Christian Hardmeier, Kellie Webster, and Will Radford, editors, *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, 2020.

Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias, October 2018.

BIG bench authors. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023.

P. Biecek. Dalex: Explainers for complex predictive models in r. *Journal of Machine Learning Research*, 19(84):1–5, 2018.

Shikha Bordia and Samuel R. Bowman. Identifying and reducing gender bias in word-level language models. In Sudipta Kar, Farah Nadeem, Laura Burdick, Greg Durrett, and Na-Rae Han, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics:*

Student Research Workshop, pages 7–15, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

Dylan Bouchard. An actionable framework for assessing bias and fairness in large language model use cases, 2024.

Confident-Ai. GitHub - confident-ai/deepeval: The LLM Evaluation Framework.

Pieter Delobelle, Ewoenam Kwaku Tokpo, Toon Calders, and Bettina Berendt. Measuring fairness with biased rulers: A survey on quantifying biases in pretrained language models. *CoRR*, abs/2112.07447, 2021.

Sunipa Dev, Tao Li, Jeff Phillips, and Vivek Srikumar. On measuring and mitigating biased inferences of word embeddings, 2019.

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*, FAccT '21, page 862–872, New York, NY, USA, 2021. Association for Computing Machinery.

Michael Feldman, Sorelle Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact, 2015.

Clémentine Fourrier, Nathan Habib, Thomas Wolf, and Lewis Tunstall. Lighteval: A lightweight framework for llm evaluation, 2023.

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, 2024.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 07 2024.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Findings*, 2020.

Giskard-Ai. Github - giskard-ai/giskard: Open-source evaluation & testing for ml models & llms.

Seraphina Goldfarb-Tarrant, Rebecca Marchant, Ricardo Muñoz Sánchez, Mugdha Pandya, and Adam Lopez. Intrinsic bias metrics do not correlate with application bias. *CoRR*, abs/2012.15859, 2020.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning, 2016.

HowieHwong. GitHub - HowieHwong/TrustGPT: Can we Trust Large Language Models?: A benchmark for responsible large language models via toxicity, Bias, and Value-alignment evaluation.

Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. Reducing sentiment bias in language models via counterfactual evaluation, 2020.

Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Hanchi Sun, Zhengliang Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bertie Vidgen, Bhavya Kailkhura, Caiming Xiong, Chaowei Xiao, Chunyuan Li, Eric P. Xing, Furong Huang, Hao Liu, Heng Ji, Hongyi Wang, Huan Zhang, Huaxiu Yao, Manolis Kellis, Marinka Zitnik, Meng Jiang, Mohit Bansal, James Zou, Jian Pei, Jian Liu, Jianfeng Gao, Jiawei Han, Jieyu Zhao, Jiliang Tang, Jindong Wang, Joaquin Vanschoren, John Mitchell, Kai Shu, Kaidi Xu, Kai-Wei Chang, Lifang He, Lifu Huang, Michael Backes, Neil Zhenqiang Gong, Philip S. Yu, Pin-Yu Chen, Quanquan Gu, Ran Xu, Rex Ying, Shuiwang Ji, Suman Jana, Tianlong Chen, Tianming Liu, Tianyi Zhou, William Yang Wang, Xiang Li, Xiangliang Zhang, Xiao Wang, Xing Xie, Xun Chen, Xuyu Wang, Yan Liu, Yanfang Ye, Yinzhi Cao, Yong Chen, and Yue Zhao. Trustllm: Trustworthiness in large language models. In *Forty-first International Conference on Machine Learning*, 2024.

Huggingface. Github - huggingface/evaluate: Evaluate: A library for easily evaluating machine learning models and datasets.

Faisal Kamiran, Asim Karim, and Xiangliang Zhang. Decision theory for discrimination-aware classification. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, ICDM '12, page 924–929, USA, 2012. IEEE Computer Society.

Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Proceedings of the 2012th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II*, ECMLPKDD'12, page 35–50, Berlin, Heidelberg, 2012. Springer-Verlag.

Katyfelkner. GitHub - katyfelkner/winoqueer.

Klara Krieg, Emilia Parada-Cabaleiro, Gertraud Medicus, Oleg Lesota, Markus Schedl, and Navid Rekabsaz. Grep-biasir: a dataset for investigating gender representation-bias in information retrieval results. In *Proceeding of the 2023 ACM SIGIR Conference On Human Information Interaction And Retrieval (CHIIR)*, 2022.

Shahar Levy, Koren Lazar, and Gabriel Stanovsky. Collecting a large-scale gender bias dataset for coreference resolution and machine translation, 2021.

Tao Li, Tushar Khot, Daniel Khashabi, Ashish Sabharwal, and Vivek Srikumar.

UnQovering stereotyping biases via underspecified questions. In *Findings of EMNLP*, 2020.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. Holistic evaluation of language models, 2023.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, Zihao Wu, Lin Zhao, Dajiang Zhu, Xiang Li, Ning Qiang, Dingang Shen, Tianming Liu, and Bao Ge. Summary of ChatGPT-related research and perspective towards the future of large language models. *Meta-Radiology*, 1(2):100017, sep 2023.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey, 2024.

Moin Nadeem, Anna Bethke, and Siva Reddy. Stereoset: Measuring stereotypical bias in pretrained language models, 2020.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Online, November 2020. Association for Computational Linguistics.

Arshaan Nazir, Thadaka Kalyan Chakravarthy, David Amore Cecchini, Thadaka Kalyan Chakravarthy, Rakshit Khajuria, Prikshit Sharma, Ali Tarik Mirik, Veysel Kocaman, and David Talby. LangTest: A comprehensive evaluation library for custom LLM and NLP models. *Software Impacts*, 19(100619), 2024.

Debora Nozza, Federico Bianchi, and Dirk Hovy. "HONEST: Measuring hurtful sentence completion in language models". In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2398–2406, Online, June 2021. Association for Computational Linguistics.

Nyu-Mll. GitHub - nyu-mll/BBQ: Repository for the Bias Benchmark for QA dataset.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics.

Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q. Weinberger. On fairness and calibration, 2017.

Partha Pratim Ray. 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, 2023.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

Pedro Saleiro, Benedict Kuester, Abby Stevens, Ari Anisfeld, Loren Hinkson, Jesse London, and Rayid Ghani. Aequitas: A bias and fairness audit toolkit. *arXiv preprint arXiv:1811.05577*, 2018.

Eric Michael Smith, Melissa Hall, Melanie Kambadur, Eleonora Presani, and Adina Williams. "i'm sorry to hear that": Finding new biases in language models with a holistic descriptor dataset. 2022.

Tensorflow. GitHub - tensorflow/fairness-indicators: Tensorflow's Fairness Evaluation and Visualization Toolkit.

Umanlp. GitHub - umanlp/RedditBias: Code & Data for the paper "RedditBias: A Real-World Resource for Bias Evaluation and Debiasing of Conversational Language Models".

Sriram Vasudevan and Krishnaram Kenthapadi. The linkedin fairness toolkit (LiFT). https://github.com/linkedin/lift, August 2020.

Vnmssnhv. GitHub - vnmssnhv/NeuTralRewriter: Neutral rewriter.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. 2023.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. Mind the gap: A balanced corpus of gendered ambiguou. In *Transactions of the ACL*, page to appear, 2018.

Hilde Weerts, Miroslav Dudík, Richard Edgar, Adrin Jalali, Roman Lutz, and Michael Madaio. Fairlearn: Assessing and Improving Fairness of AI Systems. *Journal of Machine Learning Research*, 24, 2023.

James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda B. Viégas, and Jimbo Wilson. The what-if tool: Interactive probing of machine learning models. *CoRR*, abs/1907.04135, 2019.

Wu Zekun, Sahan Bulathwela, and Adriano Soares Koshiyama. Towards auditing large language models: Improving text-based stereotype detection, 2023.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning, 2018.

Jizhi Zhang, Keqin Bao, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. Is chatgpt fair for recommendation? evaluating fairness in large language model recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, volume 2012 of *RecSys '23*, page 993–999. ACM, September 2023.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Gender Bias in Coreference Resolution: Evaluation and Debiasing methods, 4 2018.