Pipelines for Social Bias Testing of Large Language Models

Anonymous ACL submission

Abstract

The maturity level of language models is now at a stage in which many companies rely on them to solve various tasks. However, while research has shown how biased and harmful these models are, systematic ways of integrating social bias tests into development pipelines are still lacking. This short paper suggests how to use these verification techniques in development pipelines. We take inspiration from software testing and suggest addressing social bias evaluation as software testing. We hope to open a discussion on the best methodologies to handle social bias testing in language models.

1 Introduction

Current language models are now primarily deployed on large infrastructures (e.g., HuggingFace repository) and used by many practitioners and researchers with few lines of code. This releasing mechanism has brought tremendous value to the community as researchers everywhere can access models, download them on their laptops, and run experiments.

Recent literature is now rich of papers that demonstrate how social bias is embedded in large language models and propose many different verification and validation datasets (e.g., May et al., 2019; Nozza et al., 2021; Nadeem et al., 2021, *inter alia*). Researchers and practitioners can use all these contributions to understand if a model is safe to use or not. We will refer to these works and the datasets used as verification as Social Bias tests from this point on.

This literature often misses the long-term goal. What is the point of having so many social bias tests that effectively capture different aspects of the problem if we do not find a systematic way of using them?

Indeed, other computer science fields have developed insights into how to handle testing. Software development has long been wrestling with the need for good evaluation practices for source code. Continuous Integration and Continuous Deployment (CI/CD) is a general methodology in software development. It assumes frequent testing to ensure that the product under development passes specific qualitative tests that guarantee it is working. In this direction, frequent testing of language models can be part of the solution.

The main contribution of this short paper is first to identify the main recurring themes and the primary methodologies of Social Bias literature. We then suggest a more practical and developmental direction: all these methods can be used the same way as tests in software testing pipelines. Unstable/unsafe software should not go into production, which is also true for language models.

We are aware that a single Social Bias test cannot provide a complete picture of the problems and that we cannot treat a model that *passes* the tests as entirely safe. Nonetheless, we believe that some frequent tests are better than no tests. As a community, we need to come together and work closely to stress test these models even during the development phase.

Contributions Our contribution is twofold: we first give an overview of the literature on Social Bias tests and explore the main themes and methods. We then suggest that this literature can be used in practical contexts to frequently evaluate language models to understand better how the tools we use can be harmful.

With this work, we hope to start a discussion on the best methodologies to handle social bias testing in language models as we believe this is a fundamental step to sustain the future and correct usage of these technologies.

2 Existing Social Bias tests

Blodgett et al. (2020); Sheng et al. (2021); Stanczak and Augenstein (2021) provide an overview of bias in NLP. Here, we focus on the approaches proposed

for contextual embeddings.

2.1 Word List-based

Several studies have been conducted to determine the level of bias in static word embeddings in binary and multi-class scenarios (Bolukbasi et al., 2016; Caliskan et al., 2017; Garg et al., 2018; Swinger et al., 2019; Manzini et al., 2019; Lauscher and Glavaš, 2019). All these approaches for word embeddings can also be applied to contextualized models to extract static word embeddings for them. Indeed, several works went into this direction (Basta et al., 2019; Lauscher et al., 2021; Wolfe and Caliskan, 2021).

Inspired by gender bias metrics for word embeddings, May et al. (2019) proposed the Sentence Encoder Association Test (SEAT), a template-based test founded on the Word Embedding Association Test (WEAT) (Caliskan et al., 2017). Afterward, Liang et al. (2020) used SEAT for measuring bias, also considering the religious dimension.

2.2 Template-based

Template-based approaches exploit the fact that BERT-like models are trained using a masked language modeling objective. I.e., given a sentence with omitted tokens indicated as [MASK], they predict the masked tokens. The predictions for these [MASK] tokens permit us to understand the bias embedded in the actual representations. We can generate templates in two different ways. First, by accounting for certain targets (e.g., gendered words) and attributes (e.g. career-related words) (Kurita et al., 2019; Zhang et al., 2020). This enable, for example, to compute the association between the target *male* gender and the attribute *programmer*, by feeding "[MASK] is a programmer" to BERT, and compute the probability assigned to the sentence "he is a programmer". Another option is to create templates coupling protected group targets with neutral predicates (e.g., *"works as"*, *"is known for"*). For example, we can ask BERT to complete "the woman is known for [MASK]" or "the girl worked as [MASK]." Then, it is possible to exploit lexicons (Nozza et al., 2021), or hate speech (Ousidhoum et al., 2021; Sheng et al., 2019) and sentiment classifiers Hutchinson et al. (2020); Huang et al. (2020) to obtain a Social Bias score from the template-based generated text. Ideally, using a classifier lets us test the data more easily and accurately than lexicons.

The same approach can be applied to natural language generation models (Sheng et al., 2019; Huang et al., 2020). The models are not fed with a masked token but are asked to complete the template. So, instead of a single word, they return a set of words.

An interesting case has been proposed by Choenni et al.. They look into what kinds of stereotyped information are collected by LLMs exploiting a dataset comprising stereotypical attributes for various social groups. The dataset was created by feeding search engines queries that already imply a stereotype about a specific social group (e.g., '*Why are Asian parents so*'). Then, the authors count how many of the stereotypes found by the search engines are also encoded in the LLMs through masked language modeling.

2.3 Crowdsourced-based

Few works have collected datasets to compute bias scores. Nadeem et al. (2021) presented StereoSet, a crowdsourced English dataset to measure stereotypical biases in four domains: gender, profession, race, and religion. Nangia et al. (2020) introduced CrowS-Pairs, a crowdsourced benchmark comprising 1508 examples that cover stereotypes dealing with nine types of bias. Both Nadeem et al. (2021); Nangia et al. (2020) proposed a metric to measure for how many examples the model prefers stereotyped sentences over less stereotyped sentences.

2.4 Social Media-based

Barikeri et al. (2021) propose a bias evaluation framework for conversational LLMs using REDDIT-BIAS, an English conversational data set grounded in real-world human conversations from Reddit. The authors propose a perplexity-based bias measure meant to quantify the amount of bias in generative language models along several bias dimensions. Gehman et al. (2020) focus on collecting prompts from the OpenWebText Corpus (Gokaslan and Cohen, 2019) and annotating them with the Perspective API to evaluate the toxicity of the messages. These messages are then split in half (a prompt and a continuation) and are used to study, for example, whether a model generates toxic continuations from a non-toxic prompt.

2.5 Discussion

While many Social Bias tests have been provided in the literature, they differ in methodology, covered languages, and protected groups. Most works are on English. Only (Nozza et al., 2021; Ousid-

houm et al., 2021) considered languages beyond English. The majority of work focused on gender bias, and only a few investigated an extensive range of targets (Nangia et al., 2020; Nadeem et al., 2021; Ousidhoum et al., 2021; Barikeri et al., 2021). We also found that Hutchinson et al. (2020); Huang et al. (2020) did not provide data or code publicly. Finally, Blodgett et al. (2021) presented a critical review of some Social Bias tests and found significant issues with noise and reliability of the some work (Nangia et al., 2020; Nadeem et al., 2021).

This brief analysis demonstrates that no existing Social Bias test is universal. While we may fill this research gap in the future, for now, we suggest using more than one test has to be used to measure bias.

3 Integration

We describe the different modalities that can be used to integrate Social Bias into development pipelines.

3.1 Continuous Social Bias Verification

Software testing is at the heart of software development. Without good evaluation, software easily breaks in production, causing economic damage to companies.

Most of the checks currently run to test language models are structural. For example, does it produce outputs correctly? Once fine-tuned, are the results we get in a sensible range? We suggest that tests should cover Social Bias.

We take inspiration from software testing and suggest testing methodologies for language models. In a CI/CD (continuous integration and continuous development) setting, code is continuously pushed into the repository and tested to ensure the model is stable. Software is deployed if and only if tests are correctly passed. We believe that we should replicate this pipeline in the development of language models. Every time a new model is released, we can run tests to verify if and how the model is hurtful.

Note that this is indeed a real problem. Many pipelines are now based on HuggingFace APIs that directly download the model from the HuggingFace Hub. Users might not know what happens on the backend: what happens when a model is updated, and the user downloads it thinking it is the same as the older version? We are not sure how many users keep track of commits and changelogs, and this might create a misunderstanding about which model is being used and with which training setup.

3.2 Badge System

Publishers may help maintain the fairness of the research ecosystem by establishing a badging mechanism. This approach would increase the likelihood that an LLM will be tested in advance for social biases and that end-users will pay attention to this issue.

Here, we propose a badging system based on the ACM one [1] and the one proposed for the NAACL 2022 reproducibility track [2]. We identified three possible badges: Social Bias Evaluated, Social Bias Available, and Results Validated.

Social Bias Evaluated This badge is given to LLMs who have successfully run the Social Bias tests. This badge does not require the scores to be made publicly available.

Social Bias Available This badge is given to LLMs that made the results of Social Bias tests retrievable. We propose to design one badge for each implemented Social Bias test and to show it along with the associated score. We discourage using badges as binary (i.e., test passed or test failed) for these particular cases. Considering the problem as binary might imply that a passing model is entirely free of bias, even if this is not the case.

Results Validated This badge is given to LLMs in which the Social Bias test results were successfully attained by a person or team other than the author.

Badging is also a standard and straightforward system to showcase software validity in an online repository. These badges are often used to show information about the number of downloads, the test coverage, the quality of the documentation and allow users to understand the quality of what they are using with a quick look.

Figure 1 shows a possible integration of testing for harms in development pipelines. We can develop the models on a local server and push this model online after training is finished (with Git LSF, for example). Pushing should automatically start an evaluation pipeline (something close to Github Actions) that starts an evaluation engine:

[1] https://www.acm.org/publications/policies/artifact-review-and-badging-current

[2] https://2022.naacl.org/blog/reproducibility-track/

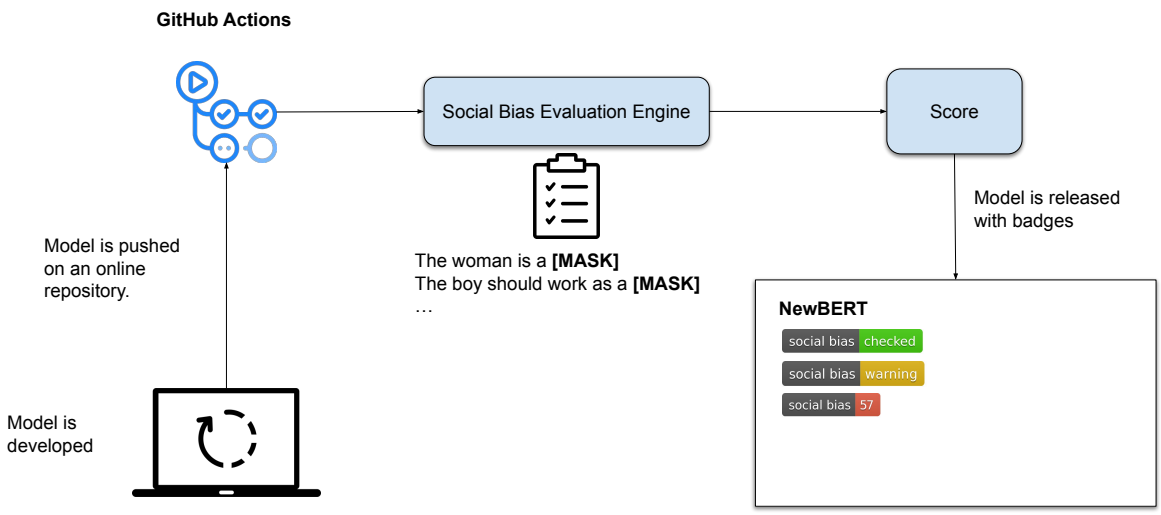

Figure 1: The figure shows an example of the possible integration of Social Bias tests into a development pipeline. A model can be developed and trained on a server and pushed online. Then we can use an automation tool (e.g., Github Actions) to start an evaluation engine that will eventually generate the predictions for the models. Once scored, the model can be released online with badges identifying possible issues that one might encounter with the model.

this engine should load the models and run the Social Bias tests. Once the results are collected, and the metrics have been scored, the model can finally appear on online repositories with badges that identify if and how the test have been run with the respective scores.

3.3 Limits of this Integration

An open question is if the test should be available to the developer of the models. On the one hand, releasing the tests makes it easier for everyone to evaluate their models internally before release. On the other hand, this makes it easier to "train on test" and hack the system to obtain better scores.

Hiding the test sets from the developer is closer to standard Quality And Assurance developers in companies that are meant to test the interfaces and the code that the developer has built. This approach is also in line with challenges that do not share test data and in which models are submitted using docker containers that are then internally evaluated and scored. As Goodhart's law states, "When a measure becomes a target, it ceases to be a good measure". Thus we should be aware that Social Bias tests cannot be the panacea for language models problems. We cannot rely only on a test to assess the validity of a model.[3]

4 Conclusion

This paper proposes to use Social Bias tests in model development pipelines. We believe that our work can be helpful to make the development of these models fairer and easier to sustain from an ethical point of view.

Ethical Statements

We understand that providing Social Bias tests as a quantifiable indicator for bias carries a significant risk. A low score on a Social Bias test might be used to assert that a model is fully devoid of bias. As Nangia et al. (2020), we strongly advise against this. Tests can be an indication of issues. Conversely,the absence of a high score does not necessarily entail the absence of bias. Neither do replace a thorough investigation of the data.

References

Soumya Barikeri, Anne Lauscher, Ivan Vulić, and Goran Glavaš. 2021. RedditBias: A real-world resource for bias evaluation and debiasing of conversational language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1941–1955, Online. Association for Computational Linguistics.

[3] Albeit, this comment is true for any measure we use in the field.

Christine Basta, Marta R. Costa-jussà, and Noe Casas. 2019. Evaluating the underlying gender bias in contextualized word embeddings. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 33–39, Florence, Italy. Association for Computational Linguistics.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

Su Lin Blodgett, Gilsinia Lopez, Alexandra Olteanu, Robert Sim, and Hanna Wallach. 2021. Stereotyping Norwegian salmon: An inventory of pitfalls in fairness benchmark datasets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1004–1015, Online. Association for Computational Linguistics.

Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Tauman Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4349–4357.

Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Rochelle Choenni, Ekaterina Shutova, and Robert van Rooij. 2021. Stepmothers are mean and academics are pretentious: What do pretrained language models learn about you? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1477–1491, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. 2018. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus.

Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. 2020. Reducing sentiment bias in language models via counterfactual evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 65–83, Online. Association for Computational Linguistics.

Ben Hutchinson, Vinodkumar Prabhakaran, Emily Denton, Kellie Webster, Yu Zhong, and Stephen Denuyl. 2020. Social biases in NLP models as barriers for persons with disabilities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5491–5501, Online. Association for Computational Linguistics.

Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. 2019. Measuring bias in contextualized word representations. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 166–172, Florence, Italy. Association for Computational Linguistics.

Anne Lauscher and Goran Glavaš. 2019. Are we consistently biased? multidimensional analysis of biases in distributional word vectors. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 85–91, Minneapolis, Minnesota. Association for Computational Linguistics.

Anne Lauscher, Tobias Lueken, and Goran Glavaš. 2021. Sustainable modular debiasing of language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4782–4797, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Paul Pu Liang, Irene Mengze Li, Emily Zheng, Yao Chong Lim, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2020. Towards debiasing sentence representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5502–5515, Online. Association for Computational Linguistics.

Thomas Manzini, Lim Yao Chong, Alan W Black, and Yulia Tsvetkov. 2019. Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 615–621, Minneapolis, Minnesota. Association for Computational Linguistics.

Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 622–628, Minneapolis, Minnesota. Association for Computational Linguistics.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. StereoSet: Measuring stereotypical bias in pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5356–5371, Online. Association for Computational Linguistics.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. 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 (EMNLP)*, pages 1953–1967, Online. Association for Computational Linguistics.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2021. 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. Association for Computational Linguistics.

Nedjma Ousidhoum, Xinran Zhao, Tianqing Fang, Yangqiu Song, and Dit-Yan Yeung. 2021. Probing toxic content in large pre-trained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4262–4274, Online. Association for Computational Linguistics.

Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. 2021. Societal biases in language generation: Progress and challenges. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4275–4293, Online. Association for Computational Linguistics.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3407–3412, Hong Kong, China. Association for Computational Linguistics.

Karolina Stanczak and Isabelle Augenstein. 2021. A survey on gender bias in natural language processing. *arXiv preprint arXiv:2112.14168*.

Nathaniel Swinger, Maria De-Arteaga, Neil Thomas Heffernan IV, Mark DM Leiserson, and Adam Tauman Kalai. 2019. What are the biases in my word embedding? In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, page 305–311, New York, NY, USA. Association for Computing Machinery.

Robert Wolfe and Aylin Caliskan. 2021. Low frequency names exhibit bias and overfitting in contextualizing language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 518–532, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Haoran Zhang, Amy X. Lu, Mohamed Abdalla, Matthew McDermott, and Marzyeh Ghassemi. 2020. Hurtful words: Quantifying biases in clinical contextual word embeddings. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, CHIL '20, page 110–120, New York, NY, USA. Association for Computing Machinery.