Automating Bias Testing of LLMs

Sergio Morales
Universitat Oberta de Catalunya
Barcelona, Spain
0000-0002-5921-9440

Robert Clarisó
Universitat Oberta de Catalunya
Barcelona, Spain
0000-0001-9639-0186

Jordi Cabot
Luxembourg Institute of Science and Technology
Esch-sur-Alzette, Luxembourg
0000-0003-2418-2489

2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE) | 979-8-3503-2996-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ASE56229.2023.00018

***Abstract*—Large Language Models (LLMs) are being quickly integrated in a myriad of software applications. This may introduce a number of biases, such as gender, age or ethnicity, in the behavior of such applications. To face this challenge, we explore the automatic generation of tests suites to assess the potential biases of an LLM. Each test is defined as a prompt used as input to the LLM and a test oracle that analyses the LLM output to detect the presence of biases.**

***Index Terms*—testing, ethics, bias, fairness, large language models**

I. Introduction

The introduction of *large language models* (LLMs), with their availability via external APIs and a growing number of open source variants[1], has facilitated the integration of generative AI features in many software applications.

As any other functionality, LLM-based features must be tested. Beyond "traditional" properties, such as accuracy, we believe a strong emphasis should be put on testing the LLM for potential biases affecting aspects such as gender, age or ethnicity. Indeed, we have seen plenty of examples where biased algorithms had harmful social consequences. For instance, back in 2015, the algorithm used for hiring candidates developed and used by Amazon was revealed discriminatory to women [1]. That same year, an independent research found that Google's advertising system was displaying higher-paying positions to men [2]. In 2019, researchers highlighted that an algorithm used in US hospitals favored white patients over black patients [3]. Another infamous case of racism was the algorithm used in US court systems to predict the likelihood that a defendant would become a recidivist [4]. LLMs are susceptible to similar problems, like the gender bias of the BERT model [5], and the racist and political bias in the earliest version of Hugging Chat [6]. Typically, LLMs are trained with datasets based on crawls from the Internet and, therefore, they often reproduce and intensify unfairness and toxicity present in the training data [7]–[10].

So far, more than 300 ethics principle sets, ontologies, or guidelines have been published in the last years to guardrail the development of AI algorithms [11]. However, there is still a gap between those guidelines and the day-to-day of developers building and integrating new AI models [12]–[15] who lack proper testing tools to assess them against fairness issues. This is a pressing concern with the growing diversity of LLMs and the ability to specialize LLMs for specific datasets, as niche LLMs may not have undergone the same level of public scrutiny as the more popular ones.

In this sense, this paper aims to provide a comprehensive test suite that developers could (a) easily adopt to assess the potential biases of LLMs, and (b) integrate in their software development projects. In particular, we contemplate the following set of fairness concerns on: gender, sexual orientation, race, age, nationality, religion, and political opinions. Our test suite combines different prompting strategies to try to uncover biases in the LLMs. Concrete prompts are derived from prompt templates instantiated with the list of key concerns the developer would like to test.

In what follows we describe our overall approach and the preliminary results, discuss the related work, and present a list of topics that are included in our future roadmap.

II. Approach

We propose a new test suite for testing ethical concerns in LLMs. Due to the nature of LLMs, we follow a black-box approach where the output of the LLM will be observed to assess the presence of biases, while the input will be a set of prompts that will try to induce the occurrence of such biases.

A. Prompt and oracle strategies for the unit testing of LLMs

Our test suite is composed by a set of prompts and the corresponding oracle evaluating whether the LLM output for such prompt satisfies the oracle condition in charge of assessing the presence or absence of bias.

We have followed different strategies to define the prompts, inspired by examples of well-known bias issues raised by the media over the last years. Some prompts are specialized in triggering a concrete bias, while others can be easily adapted to many biases (see also next section II-B)[2]. The main strategies are the following:

- Direct questions or statements that the LLM is asked to confirm or deny, *e.g.*, *"Are black people inferior to white people?"*; and
- Queries to be completed, where the LLM output might show a biased answer or an unbalanced classification,

This work has been partially funded by the Spanish government (PID2020-114615RB-I00/AEI/10.13039/501100011033, project LOCOSS); the Luxembourg National Research Fund (FNR) PEARL program, grant agreement 16544475; and the AIDOaRt project, which has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 101007350.

[1]https://github.com/Hannibal046/Awesome-LLM#open-llm

[2]In the test suite, prompts are labelled with the different concerns that each is addressing, which facilitates their selection depending on the testing goal.

2643-1572/23/$31.00 ©2023 IEEE
DOI 10.1109/ASE56229.2023.00018

e.g., *"List the 3 most probable skin color for a bartender, and note next to each skin color its probability"*; and
- Prompt sets where all prompts are equal except for a specific characteristic (gender, age, ethnicity,... that takes different values on each one) and for which the LLM may provide a different answer.

Note that each strategy could be enriched to look for biases appearing only when mentioning a combination of characteristics, *e.g.*, a model that avoids gender biases may exhibit them when gender with age are combined.

Each prompt strategy requires a different type of *oracle*. For direct questions, the oracle needs to evaluate whether the LLM is giving a positive or negative answer. For completions, we should check whether the answer follows the stereotypes we are trying to avoid. Finally, for the set prompting strategy, we may consider differences in the content or the positive/negative tone of the answers. In all cases, the oracle could be evaluated via a simple analysis of the textual response (*e.g.*, looking for the occurrence of certain keywords or the Levenshtein distance between two strings). Alternatively, we could use a second LLM model, trained to detect biases [16], to evaluate the first one's response. For the first strategy is enough to ask whether the answer is positive or negative. For the other two, we would need to ask this second LLM whether the response (or the difference between the two responses) is biased. Obviously, as soon as a second LLM is introduced, there is now a second non-deterministic element in the testing pipeline. This, using an LLM as the oracle, is again a trade-off.

In any case, beyond the oracle decision, we provide the evaluator with annotated examples of distorted outputs for an additional manual evaluation if so desired.

B. Automatic test generation

Each of the above prompt strategies can be regarded as a prompt template where parts of the prompts will be instantiated based on the main interests of the tester. For instance, the first example above could be reformulated as the following prompt template: *"Are {SKIN_COLOR1} people inferior to {SKIN_COLOR2} people?"*; and then instantiated with the desired skin color combinations.

In some scenarios, it may be desirable to test them all combinations. However, in others it may be better to focus on certain ones (depending on the profile of the users) to maximize the confidence of the results in that specific group while keeping the testing suite size manageable.

III. Related Work

LLMs are already being brought forward to generate testing code for software [17], [18], but less attention has been devoted to the challenge of testing the LLMs themselves.

Some works have focused on detecting or declaring biases on datasets [19], [20], which may instill biases in ML models trained with them. Our approach enables assessing whether those biases are present or not.

Other attempts to detect biases in LLMs specialize in gender. Zhao et al. [21] test whether LLMs favor one gender over the other when completing sentences about certain job occupations stereotypically linked to one gender. Dhamala et al. [22] and Alnegheimish et al. [23] use sentences describing specific organizations or occupations from Wikipedia as prompts to evaluate the probability of appearance of gendered pronouns and/or text with positive/negative connotations in the text generated afterward. We contribute a more flexible strategy where developers have more control on the prompts (and corresponding biases) they want to check and a richer detection mechanism by proposing different prompt strategies (such as direct questions) and not just sentence completion. An alternative approach, by Schick et al. [24], explicitly adds instructions in the prompt to ask the LLM not to generate a biased response. This could be useful in some scenarios, but it requires users to remember to add such additional instructions and trust that the LLM will be able to properly follow them.

BiasAsker [25] uses combined annotated properties to query conversational AIs. However, its number of prompt categories is limited to 3, whereas we allow introducing further strategies.

To sum up, we aim for an adaptable solution covering a larger number of biases with a combination of different prompting strategies, plus other relevant features described in the next section as part of our roadmap.

IV. Conclusions and Further Work

We have presented our first steps towards the automatic testing of harmful biases in LLMs. There are several directions in which we plan to continue exploring this topic:

- Detecting biases in text-to-image and text-to-video generators, with the challenge of developing oracles able to detect a potential bias in these types of media outputs.
- Adding tolerance levels to the testing process. Ideally, LLMs should have zero bias, but this may imply trade-offs (*e.g.*, in terms of the quality and quantity of data, the training costs,...) that an organization may not want to assume. Defining a tolerance level would imply that tests can pass if only a certain number (or degree) of biases are detected. Note that the non-determinism of LLMs will also play an important factor here.
- Testing hidden biases with deeper conversations. LLMs are becoming better at avoiding biases with simple prompts, but biases could still be revealed as part of a conversation involving a series of prompts forcing the LLM to iterate on previous responses.
- Extending our test suite to cover additional ethical and fairness concerns, for instance immoral [26] or unlawful recommendations.
- Exploring the generation of domain- or application-specific fairness tests.
- On the tooling side, our plan is to grow, with the help of the community, the number and variety of prompts for each type of bias and systematically test existing LLMs to increase the awareness of this problem. A monitoring dashboard displaying an overview of the health of LLM with respect to the different biases will also be provided.

REFERENCES

[1] "Amazon scraps secret AI recruiting tool that showed bias against women," https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G, accessed 25 May 2023.

[2] "Discriminating algorithms: 5 times AI showed prejudice," https://www.newscientist.com/article/2166207-discriminating-algorithms-5-times-ai-showed-prejudice, accessed 25 May 2023.

[3] "Racial Bias Found in a Major Health Care Risk Algorithm," https://www.scientificamerican.com/article/racial-bias-found-in-a-major-health-care-risk-algorithm, accessed 25 May 2023.

[4] "How We Analyzed the COMPAS Recidivism Algorithm," https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm, accessed 25 May 2023.

[5] R. Bhardwaj, N. Majumder, and S. Poria, "Investigating gender bias in BERT," *Cognitive Computation*, vol. 13, no. 4, pp. 1008–1018, 2021.

[6] "Hugging Face releases its own version of ChatGPT," https://techcrunch.com/2023/04/25/hugging-face-releases-its-own-version-of-chatgpt, accessed 25 May 2023.

[7] C. Basta, M. R. Costa-jussà, and N. Casas, "Evaluating the Underlying Gender Bias in Contextualized Word Embeddings," in *Proceedings of the First Workshop on Gender Bias in NLP*. Association for Computational Linguistics, Aug. 2019, pp. 33–39.

[8] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, "Man is to computer programmer as woman is to homemaker? Debiasing word embeddings," *Advances in neural information processing systems*, vol. 29, 2016.

[9] S. Gehman, S. Gururangan, M. Sap, Y. Choi, and N. A. Smith, "RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models," in *EMNLP*. Association for Computational Linguistics, Nov. 2020, pp. 3356–3369.

[10] E. Sheng, K.-W. Chang, P. Natarajan, and N. Peng, "The Woman Worked as a Babysitter: On Biases in Language Generation," in *EMNLP-IJCNLP*. Association for Computational Linguistics, Nov. 2019, pp. 3407–3412.

[11] A. Harrison, D. Spagnuelo, and I. Tiddi, "An ontology for ethical AI principles," *Semantic Web Journal*, 2021.

[12] J. C. Ibáñez and M. V. Olmeda, "Operationalising AI ethics: how are companies bridging the gap between practice and principles? An exploratory study," *AI & SOCIETY*, vol. 37, no. 4, pp. 1663–1687, 2022.

[13] T. Hagendorff, "The ethics of AI ethics: An evaluation of guidelines," *Minds and machines*, vol. 30, no. 1, pp. 99–120, 2020.

[14] Q. Lu, L. Zhu, X. Xu, J. Whittle, D. Douglas, and C. Sanderson, "Software Engineering for Responsible AI: An Empirical Study and Operationalised Patterns," in *ICSE-SEIP*. ACM, 2022, p. 241–242.

[15] J. Morley, L. Kinsey, A. Elhalal, F. Garcia, M. Ziosi, and L. Floridi, "Operationalising AI ethics: barriers, enablers and next steps," *AI & SOCIETY*, pp. 1–13, 2021.

[16] S. Prabhumoye, R. Kocielnik, M. Shoeybi, A. Anandkumar, and B. Catanzaro, "Few-shot instruction prompts for pretrained language models to detect social biases," *arXiv preprint arXiv:2112.07868*, 2021.

[17] M. L. Siddiq, J. Santos, R. Hasan Tanvir, N. Ulfat, F. Al Rifat, and V. Carvalho Lopes, "Exploring the Effectiveness of Large Language Models in Generating Unit Tests," *arXiv e-prints*, pp. arXiv–2305, 2023.

[18] Z. Yuan, Y. Lou, M. Liu, S. Ding, K. Wang, Y. Chen, and X. Peng, "No More Manual Tests? Evaluating and Improving ChatGPT for Unit Test Generation," *arXiv preprint arXiv:2305.04207*, 2023.

[19] J. Giner-Miguelez, A. Gómez, and J. Cabot, "DescribeML: A Tool for Describing Machine Learning Datasets," in *MODELS*. ACM, 2022, p. 22–26.

[20] A. Yohannis and D. Kolovos, "Towards Model-Based Bias Mitigation in Machine Learning," in *MODELS*. ACM, 2022, p. 143–153.

[21] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang, "Gender bias in coreference resolution: Evaluation and debiasing methods," *arXiv preprint arXiv:1804.06876*, 2018.

[22] J. Dhamala, T. Sun, V. Kumar, S. Krishna, Y. Pruksachatkun, K.-W. Chang, and R. Gupta, "Bold: Dataset and metrics for measuring biases in open-ended language generation," in *ACM conference on fairness, accountability, and transparency*, 2021, pp. 862–872.

[23] S. Alnegheimish, A. Guo, and Y. Sun, "Using natural sentence prompts for understanding biases in language models," in *"Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies"*. "Association for Computational Linguistics", Jul. 2022, pp. 2824–2830.

[24] T. Schick, S. Udupa, and H. Schütze, "Self-Diagnosis and Self-Debiasing: A Proposal for Reducing Corpus-Based Bias in NLP," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1408–1424, 12 2021.

[25] Y. Wan, W. Wang, P. He, J. Gu, H. Bai, and M. Lyu, "BiasAsker: Measuring the Bias in Conversational AI System," *arXiv preprint arXiv:2305.12434*, 2023.

[26] P. Ma, Z. Li, A. Sun, and S. Wang, "Oops, Did I Just Say That? Testing and Repairing Unethical Suggestions of Large Language Models with Suggest-Critique-Reflect Process," *arXiv preprint arXiv:2305.02626*, 2023.