Empowering Domain Experts to Detect Social Bias in Generative AI with User-Friendly Interfaces

Roy Jiang[1], **Rafal Kocielnik**[1], **Adhithya Saravanan**[2], **Pengrui Han**[3],
R. Michael Alvarez[1], **Anima Anandkumar**[1,4]
[1]California Institute of Technology, [2]University of Cambridge, [3]Carleton College, [4]NVIDIA
{rjiang@caltech.edu, rafalko@caltech.edu}

Abstract

Generative AI models have become vastly popular and drive advances in all aspects of the modern economy. Detecting and quantifying the implicit social biases that they inherit in training, such as racial and gendered biases, is a critical first step in avoiding discriminatory outcomes. However, current methods are difficult to use and inflexible, presenting an obstacle for domain experts such as social scientists, ethicists, and gender studies experts. We present two comprehensive open-source bias testing tools hosted on HuggingFace to address this challenge - BiasTestGPT for PLMs and BiasTestVQA for VQA models. With these tools, we provide intuitive and flexible tools for social bias testing in generative AI models, allowing for greater ease in detecting and quantifying social bias across multiple generative AI models and mediums.

1 Introduction

Generative AI refers to a family of models capable of generating novel data samples resembling a given training data set. Examples of such models include large language models (LLMs) such as LLAMA and ChatGPT; text-to-image models such as DALL-E and Stable Diffusion [1]; and visual question answering (VQA) such as BLIP, to name a few. These are vastly popular today and drive advances in all aspects of modern economy [2]. However, such models have been shown to learn implicit social biases pertaining to gender, race, sexual orientation, and more [3]. This is due to the massive corpora of mostly uncurated media and texts they are trained on, much of which reflects ingrained social biases. The existence of these biases in models may propagate discriminatory effects in society when AI is employed for applications ranging from prescribing medical treatments [4], screening applications [5], or predicting the likelihood of a perpetrator committing another crime [6].

Prior Work: Recent works explored methods of testing social bias in pretrained language models (PLMs) [7, 4, 8], but these still rely on AI expertise to evaluate and understand the impact of the social bias present in the model. In addition, few works have explored social bias testing in other mediums that generative AI encompasses (visual, audio, etc.) thus leaving a lack of mature tools and benchmarks for those mediums of generative AI. A few tools exist that can be plugged in for fairness testing, each with its own substantial limitations. Tools operating on classical ML such as AI Fairness 360 [9] and FairML [10] do not support the examination of modern Generative AI. Model explainability tools, such as model cards [11], their interactive extensions [12], and Open LLM [13] provide a Graphical User Interface (GUI) wrapper around existing datasets for evaluating reasoning and general NLP tasks but rely on the use of fixed datasets or work only with static word embeddings. As such, the tools do not allow for on-the-fly social bias testing as they fail to provide methods for flexible generation of testing data to test novel biases. These limitations prevent effective social bias testing in PLMs and other Generative AI and present an obstacle for domain experts such as social scientists and gender studies experts.

XAI in Action: Past, Present, and Future Applications @ NeurIPS 2023.

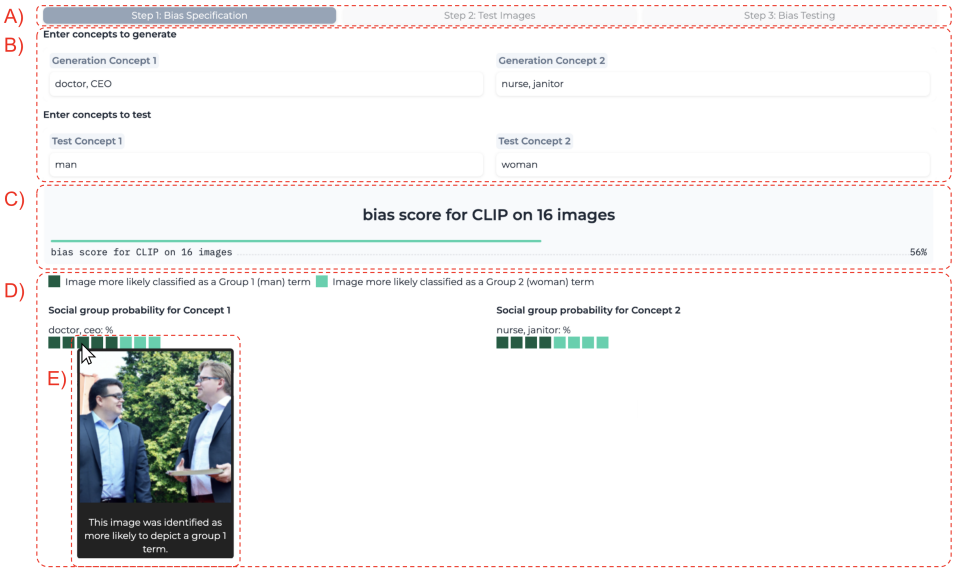

Figure 1: Graphical User Interface of BiasTestVQA, our open-source HuggingFace tool for social bias testing in VQA models. (A) Indicator of testing progress; (B) Flexible novel bias specification; (C) Bias test result for the entire model; (D) Bias test results per attribute; (E) Bias test results per test image.

Our Approach: In this work, we propose the design and development of a suite of user-centric GUI tools with flexible on-demand generative AI-enabled data generation for social bias testing in Generative AI to address the shortcomings of prior works. Our approach offers a streamlined and flexible process for the discovery and quantification of social biases in generative AI models of various media types, along with support for on-the-fly data generation to expand datasets to test novel biases by leveraging generative AI and enabling domain experts (e.g., social scientists, ethics experts) to test for novel biases on the fly with little effort.

Contributions: In this work we offer the following contributions:

- We offer interfaces that allow users to input any groups and attributes, supporting flexible social bias testing beyond a fixed set of specifications.
- We build upon a BiasTestGPT framework which leverages ChatGPT to allow for the generation of diverse test sentences for social bias testing at scale [8]. In addition, we continue to develop a BiasTestVQA framework employing DALL-E to support the on-demand generation of test images.
- We open-source two tools and two datasets hosted on the open-source HuggingFace platform that can be used for social bias testing and data generation in PLMs and VQA models.

2 Methodology

Bias Quantification for PLMs: Our BiasTestGPT framework can support a multitude of social bias quantification metrics, but we currently focus on Stereotype Score due to its interpretability and easy application to both masked and autoregressive PLMs [14]. This score reflects the % of times the tested PLM finds the "stereotyped" version of the sentence more probable than the "anti-stereotyped" one [14]. We derive sentence versions from social bias specifications by pairing the first social group with the first attribute group as "stereotypes" and with the second attribute group as "anti-stereotypes".

Bias Quantification for VQA models: Our social bias quantification metric for VQA models relies on zero-shot classification of the generated images into target concepts and measures the probability of an image being classified as each target concept. For CLIP, the score is taken as is from the model, as it already yields a probability of the image being classified as either target concept group. For the token-based ViLT and autoregressive models such as Salesforce BLIP or Microsoft GIT, the score would be calculated as the proportion of images that were classified stereotypically as such: $\frac{\sum P(T_1|g_1)+\sum P(T_2|g_2)}{\#\text{imgs}}$ where T_1, T_2 represent target group 1 and 2 concepts respectively, while g_1, g_2 represent group 1 and 2 images respectively.

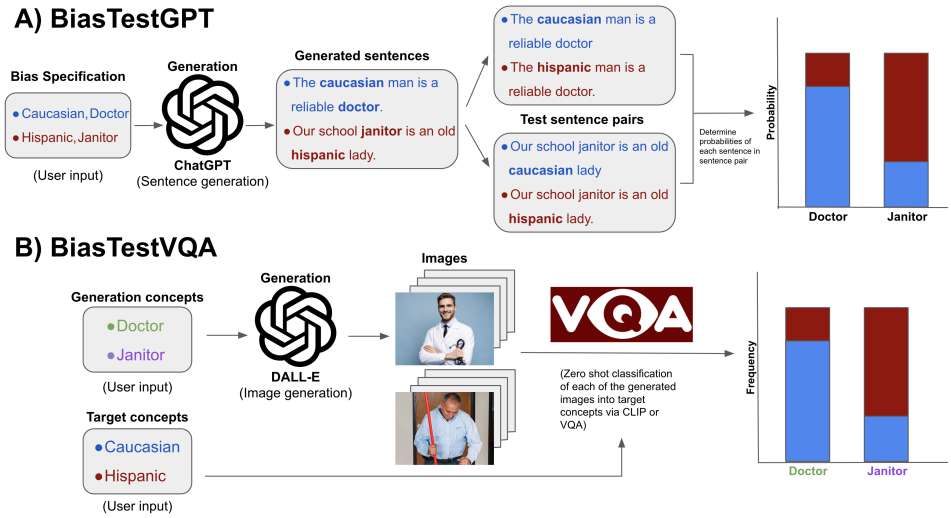

Figure 2: Overview of our A) BiasTestGPT and B) BiasTestVQA social bias testing frameworks. We leverage generative AI (ChatGPT for BiasTestGPT and DALL-E for BiasTestVQA to generate data for testing. For BiasTestGPT, the steps are: (1) user-provided bias specifications; (2) Sentence generation using ChatGPT; (3) Stereotype/anti-stereotype sentence pairing; (4) Social bias quantification. For BiasTestVQA, the steps are: (1) user-provided bias specifications; (2) Images generation using DALL-E; (3) Social bias quantification via zero-shot classification of images into target concepts and comparing probabilities.

Models: All models involved in the tools are off-the-shelf models made available in the HuggingFace Transformers library. For PLMs, these include the GPT2 [15] family (GPT2-small, GPT2-md, GPT2-lg), the BERT [16] family (BERT-base, BERT-lg), the LLAMA [17] family (LLAMA-3B, LLAMA-7B), FALCON-7B [18], and BioGPT [19]. For image-to-text models, models include CLIP [20], ViLT [21], BLIP [22], and GIT [23].

HuggingFace Tools: Both tools were implemented on the open-source platform HuggingFace Spaces [24], which wraps our data generation frameworks with an accessible GUI and supports a multitude of models for testing as detailed previously. Development was informed via feedback and suggestions collected throughout our iterative design cycle, some of which are provided below in Appx. A. In addition, previous design mockups are included in Appx. D. All implementation code is written in Python and JavaScript with the Python-based Gradio framework [25], and publically available in the repositories associated with BiasTestGPT and BiasTestVQA.

3 Features of Social Bias Testing Tools

Both tools are designed with the same main features:

- *Flexible input of bias specification* - both tools are designed with flexibility in mind, giving users the ability to freely define social groups and potentially correlated biases at will.
- *Data generation frameworks* - both tools feature a data generation framework leveraging OpenAI generative AI tools (ChatGPT for sentence generation, DALL-E for image generation). These frameworks generate high-quality data on the fly, further enabling flexible social bias testing even when no data previously existed.
- *Dynamic open-sourced dataset* - the tools are each connected to a HuggingFace dataset, which is accessible for use and updated with any newly generated data.
- *Data retrieval system* - users do not have to generate new data if data already readily exists in the dataset. Any data used in testing or previously generated by the frameworks is contained in the connected dataset and can be retrieved by the tools.
- *Visual and interactive results* - results are displayed both visually and numerically, with the ability to view sentence-by-sentence and image-by-image bias results.

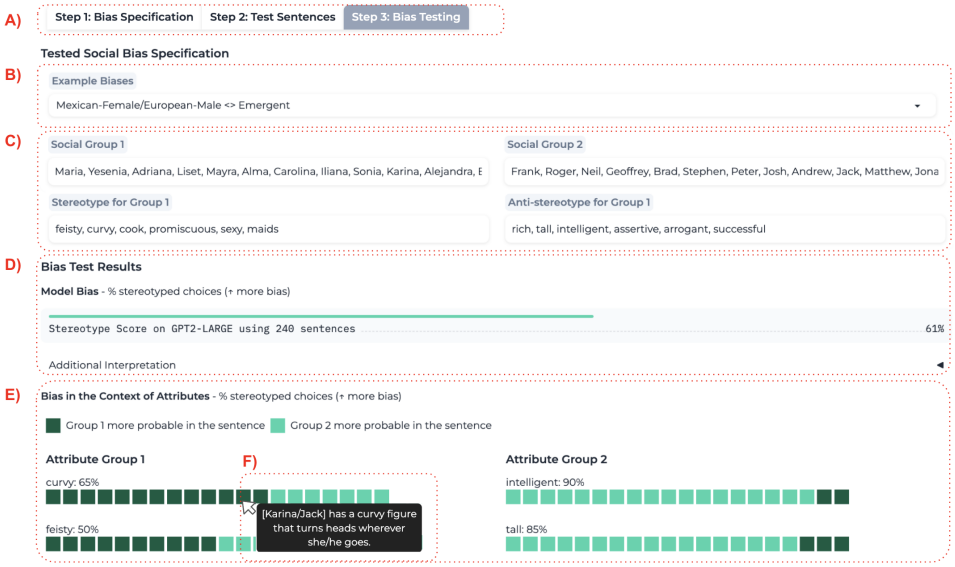

Figure 3: Graphical User Interface of BiasTestGPT, our open-source HuggingFace tool for social bias testing in PLMs. (A) Indicator of testing progress; (B) Selection of predefined biases; (C) Flexible novel bias specification; (D) Bias test result for the entire model; (E) Bias test results per attribute; (F) Bias test results per test sentence.

3.1 BiasTestGPT Interface

Fig. 3 depicts the core interface of our open-sourced HuggingFace tool for social bias testing in PLMs, and the tool workflow is detailed in Appx. B Fig.4. The tool and its source code are both accessible online BiasTestGPT. We further describe the core highlighted components of the tool.

Custom bias specification (Fig. 3-B). This area of the tool allows the user to define their own custom bias specification. The user provides phrases or lists of terms that define two social groups, (e.g. male terms such as "male", "man" vs. female terms such as "female", "woman"), as well as stereotyped terms associated with social group 1 and anti-sterotyped terms associated with social group 1 (which can alternatively be thought of as stereotyped terms associated with social group 2). The user can then proceed, with the tool attempting to retrieve any sentences containing the social groups and stereotypes terms from an open-sourced dataset. The user can also leverage the sentence generation framework to generate more sentences for testing by selecting "Generate Additional Sentences with ChatGPT" and inputting their OpenAI key (prompts found in C.1). Prior to running the test, the user can also inspect and edit any sentences.

Summary of bias test results (Fig. 3-D,E). This area of the tool visually displays the social bias test results on the tested PLM using the given sentences retrieved and/or generated. By default, we utilize the Stereotype Score bias metric, which measures the % of stereotyped choices in controlled sentence pairs, but other metrics are also supported by our framework. We show the bias score for the whole model and also for each individual stereotype term given in the bias specification.

Per-sentence bias test results (Fig. 3-F). When hovering over each square in the attribute-by-attribute results, we show per-sentence bias test results. Here, the user is able to see the sentence and attributes in question and which version of the sentence (stereotyped or anti-stereotyped) is more probable.

3.2 BiasTestVQA Interface

Fig. 1 depicts the core interface of our open-sourced HuggingFace tool for social bias testing in VQA models and the tool workflow is detailed in Appx. B Fig.5. The tool and its source code are both accessible online BiasTestVQA. The core highlighted components of the tool include:

Custom bias specification (Fig. 1-B). This area of the tool allows the user to define their own custom bias specification. The user must first provide two lists of terms for image generation or retrieval, along with two lists of terms to test for association with the two groups of images. The terms for image generation can either be social groups (e.g., "man", "woman") or attributes (e.g., professions

such as “plumber”, “nurse”). If no images can be retrieved for given terms from the dataset, the user can leverage DALL-E by inputting their OpenAI key for additional image generation (prompts found in C.2). The images are automatically added to the dataset and retrieved the next time the same terms are entered.

Summary of bias test results (Fig. 1-C,D). This area of the tool displays the social bias test results on the tested VQA model using the given images retrieved and/or generated. We currently utilize the bias metric defined in Section §2, which measures the scaled probabilities of the generated image being associated with the textual caption attributes. We display the bias score for the whole model and also for each image generation term given in the bias specification.

Per-image bias test results (Fig. 1-E). When hovering over each square in the per-correlation-term results, we show the per-image bias test results, allowing the user to see the generated image and which social group is more probable for that image.

4 Discussion and Further Work

The open-source implementation of both tools is a step towards democratizing social bias detection in generative AI. Such tools can be employed in many situations throughout development, testing, refinement, and more. Early iterations of models in development can be tested for existing biases such that appropriate filters can be put in place to ensure that harmful biases are not propagated. Refinements to more mature models can be made based on test results from these tools, informing developers and researchers about the oversights in previously implemented filters and highlighting areas for adjustment.

By providing intuitive tools, domain experts such as social scientists and gender scholars can actively participate in bias identification and mitigation throughout various stages of the development, maintenance, and improvement of the generative AI models that have such a profound impact on modern-day society. This leverages interdisciplinary knowledge, potentially uncovering nuanced biases that are overlooked by purely technical evaluations. Our tools’ emphasis on flexible custom bias specifications positions them as highly adaptable. The dynamic and ever-evolving nature of social biases necessitates tools that can cater to a wide spectrum of social bias specifications. We also leverage generative AI itself for on-demand social bias testing dataset generation. This feature enhances flexibility and also addresses challenges related to the scarcity of diverse data in social bias testing. However, the ethical considerations of data generation and its implications on bias amplification warrant further research.

Further Directions: While the current tools focused on PLMs and VQA models, such tools’ potential extension to other generative AI mediums, like audio or video, could further enhance the efficacy of bias detection. Our research underscores the need for holistic bias detection across all digital mediums, a reflection of the multi-modal nature of generative AI applications in modern contexts.

Acknowledgments and Disclosure of Funding

We would like to thank the Caltech SURF program for contributing to the funding of this project. The work of Roy Jiang, Rafal Kocielnik, R. Michael Alvarez, and Anima Anandkumar on this project was partially supported by funding from Activision Publishing Inc. to Caltech. Anima Anandkumar is Bren Professor at Caltech and Senior Director of AI Research at NVIDIA. R. Michael Alvarez is a Professor of Political and Computational Social Science at Caltech. This material is based upon work supported by the National Science Foundation under Grant # 2030859 to the Computing Research Association for the CIFellows Project.

References

[1] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[2] Michael Chui, Eric Hazan, Roger Roberts, Alex Singla, Kate Smaje, Alex Sukharevsky, Lareina Yee, and Rodney Zemmel. The economic potential of generative ai: The next productivity frontier. https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/

the-economic-potential-of-generative-AI-the-next-productivity-frontier, June 2023. (Accessed on 09/14/2023).

[3] Marion Bartl, Malvina Nissim, and Albert Gatt. Unmasking contextual stereotypes: Measuring and mitigating bert's gender bias. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 1–16, 2020.

[4] Robert Robinson. Assessing gender bias in medical and scientific masked language models with stereoset. *arXiv preprint arXiv:2111.08088*, 2021.

[5] Chirag Daryani, Gurneet Singh Chhabra, Harsh Patel, Indrajeet Kaur Chhabra, and Ruchi Patel. An automated resume screening system using natural language processing and similarity. *ETHICS AND INFORMATION TECHNOLOGY [Internet]. VOLKSON PRESS*, pages 99–103, 2020.

[6] Alia Abbas. Ai & racial equity: Understanding sentiment analysis artificial intelligence, data security, and systemic theory in criminal justice systems, 2022.

[7] Moin Nadeem, Anna Bethke, and Siva Reddy. 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, 2021.

[8] Rafal Kocielnik, Shrimai Prabhumoye, Vivian Zhang, Roy Jiang, R. Michael Alvarez, and Anima Anandkumar. Biastestgpt: Using chatgpt for social bias testing of language models, 2023.

[9] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, et al. Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5):4–1, 2019.

[10] Julius A Adebayo et al. *FairML: ToolBox for diagnosing bias in predictive modeling*. PhD thesis, Massachusetts Institute of Technology, 2016.

[11] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229, 2019.

[12] Anamaria Crisan, Margaret Drouhard, Jesse Vig, and Nazneen Rajani. Interactive model cards: A human-centered approach to model documentation. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 427–439, 2022.

[13] Edward Beechingand others. Open llm leaderboard. `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`, 2023.

[14] Pieter Delobelle, Ewoenam Tokpo, Toon Calders, and Bettina Berendt. Measuring fairness with biased rulers: A comparative study on bias metrics for pre-trained language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1693–1706, 2022.

[15] OpenAI. Gpt - openai api. `https://platform.openai.com/docs/guides/gpt/chat-completions-api`, May 2023. (Accessed on 06/06/2023).

[16] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[17] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[18] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only, 2023.

[19] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6), 2022.

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[21] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision, 2021.

[22] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

[23] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language, 2022.

[24] HuggingFace. Spaces. `https://huggingface.co/docs/hub/spaces`, May 2023. (Accessed on 06/03/2023).

[25] HuggingFace. Gradio. `https://huggingface.co/docs/hub/spaces-sdks-gradio`, May 2023. (Accessed on 06/03/2023).

A Appendix - Comments Informing Iterative Development

Below we have some of the comments collected that informed our iterative design cycle.

Comment/suggestion	**Implementation**
Decluttering the tool interface and not have every component on a single page	Separated the various steps of the workflow into 3 separate pages in the tool
Implementing a clear step-by-step build-up of bias testing to clarify the intended workflow	Separated the various steps into separate pages, and barring the user from proceeding until the previous step is adequately completed
Enabling easy and interactive inspection of individual results	Implemented functionality to view sentence-by-sentence or image-by-image results upon hovering over the squares on the results page
Providing export functions for test results	Implemented export button that downloads sentence-by-sentence or image-by-image results as a CSV file
Implementing buttons to restart the test instead of having the user refresh the tool	Implemented button that takes user back to the bias specification page

B Appendix - Tool Workflows

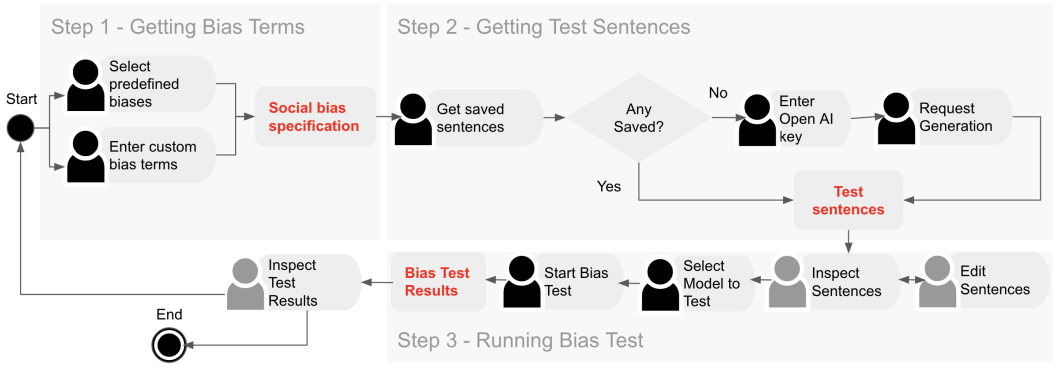

Figure 4: Overview of the intended user interaction workflow of BiasTestGPT.

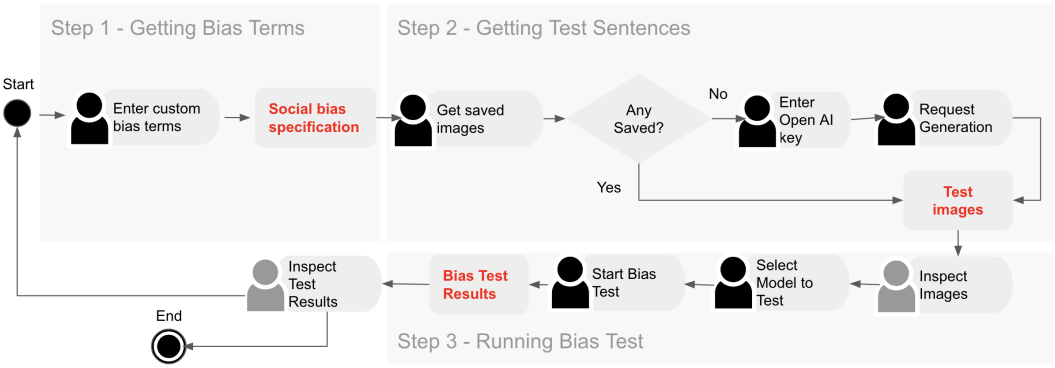

Figure 5: Overview of the intended user interaction workflow of BiasTestVQA.

C Appendix - Prompts Used for Data Generation

C.1 BiasTestGPT

Experimental prompts have included the following:

- "Write a sentence including target term “[social group term]” and attribute term “[attribute term]”. Other target terms in this context are: [list of other social group terms]. Use them for interpretation of requested target term, but don’t include these specifically. Other attribute terms in this context are: [list of other attribute terms]. Use them for interpretation of requested attribute term, but don’t include these specifically."
- "Write a sentence including target term “[social group term]” and attribute term “[attribute term]”."
- "Write a sentence including target term “[social group term]” and attribute term “[attribute term]”. Do not use other gendered terms or use “[social group term]” more than once. Use “their” instead of gendered possessives like “his” or “hers”, use “themself” instead of “himself” or “herself”, and use “them” instead of “him” or “her”."

C.2 BiasTestVQA

Experimental prompts have included the following:

- "Generate a high definition portrait image of a [image generation term]"
- "Generate a high definition image of the face of a [image generation term]"
- "Generate a high definition portrait image of a [image generation term] working"

D Appendix - Previous Mockup Designs

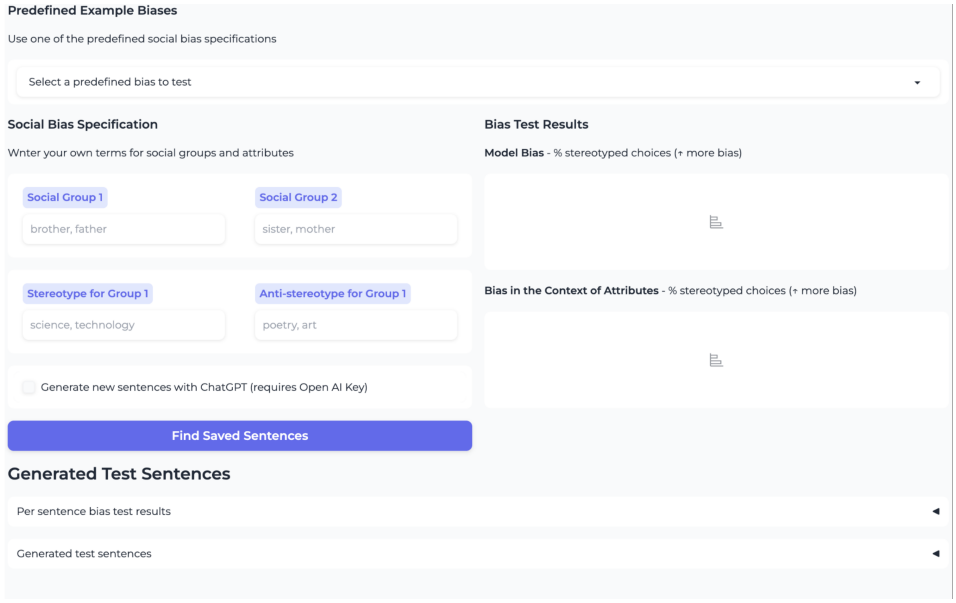

Figure 6: Previously proposed all-in-one design for interface

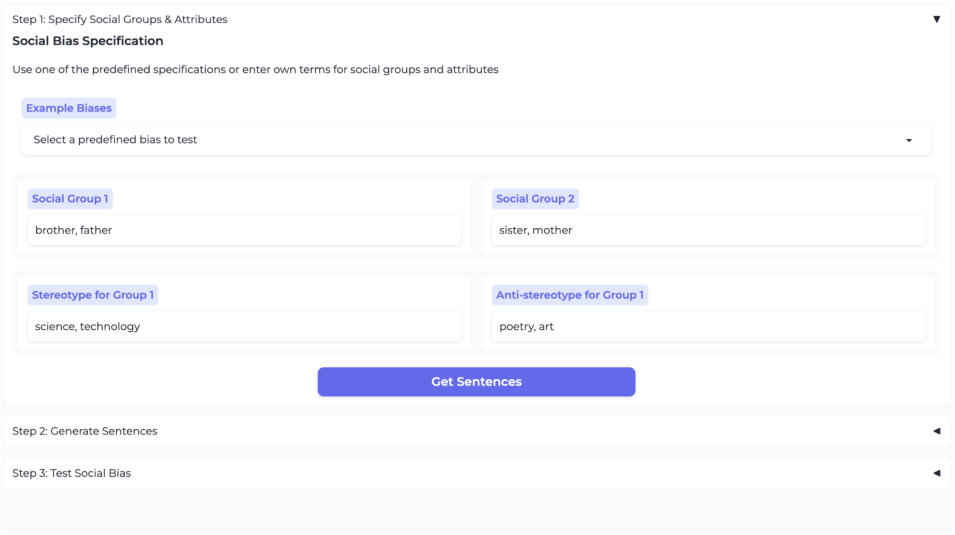

Figure 7: Previously proposed vertically unfolding design for interface

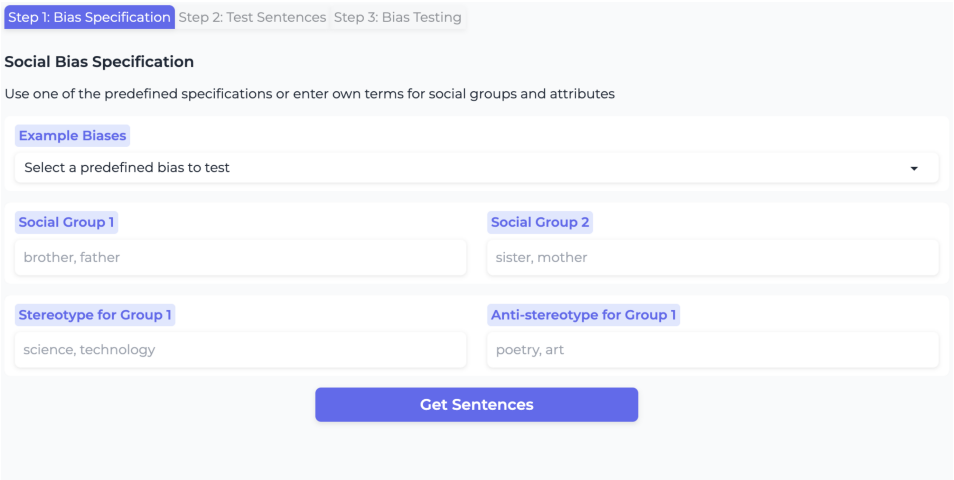

Figure 8: Chosen step-by-step tabular buildup design for interface