BiasHeal: On-the-Fly Black-Box Healing of Bias in Sentiment Analysis Systems

***Abstract*—Although Sentiment Analysis (SA) is widely applied in many domains, existing research has revealed that the unfairness in the SA prediction systems can be harmful to the welfare of less privileged people. Several works have proposed pre-processing and in-processing methods to eliminate bias in SA systems, but little attention is paid to utilizing post-processing methods to heal bias. Post-processing methods are particularly important for systems that use third-party SA services. Systems that use such services have no access to the SA engine or its training data and thus cannot apply pre-processing or in-processing methods. Therefore, this paper proposes a black-box post-processing method to make an SA system heal bias and construct fair results when bias is detected. We propose and investigate 6 self-healing strategies. Our evaluation results on two datasets show that the best strategy can construct fair results and improve accuracy on the two datasets by 2.76% and 2.85%, respectively. To the best of our knowledge, our work is the first self-healing method that can be deployed to ensure SA fairness without requiring access to the SA engine or its training data.**

***Index Terms*—Software Fairness, Sentiment Analysis, Bias Healing**

I. Introduction

Sentiment Analysis (SA) systems are widely used in modern life. But existing research [1], [2] indicates that SA systems, just like other machine learning systems, can be biased towards sensitive attributes, e.g., gender, race, occupation, and so on, which may damage the welfare of people unexpectedly. Such unfairness obviously violates various ethical standards, e.g., the European Union's ethical requirements for trustworthy AI systems [3], and needs to be carefully addressed.

Fairness has been viewed as an essential aspect of software quality [4] and researchers have invested efforts to deal with bias in SA systems. Prior efforts can be divided into two families: *bias detection* and *bias elimination*. For bias detection, many prior works [5], [6], [7] have made use of the following fundamental metamorphic relationship: a fair SA system should have the same prediction for a pair of texts that only differ in words reflecting gender information, e.g., names and pronouns. Researchers firstly produce templates by masking gender-related words with placeholders. Then, by replacing these placeholders with different gender information to generate text mutants, a number of test cases for revealing bias can be obtained. CheckList [6] and EEC [5] predefine some short text templates and generate mutants by replacing placeholders with different names and tokens that indicate genders. ASTRAEA [2] utilizes context-free grammar to generate flexible templates and more mutants. We call them static template approaches. Different from the above, BiasFinder [1] adopts a dynamic template approach, extracting names and pronouns in a text corpus to automatically and dynamically create new templates and then generating mutants from the templates.

Bias elimination methods can be divided into three types: pre-processing, in-processing, and post-processing. Pre-processing methods, which are run before training the models, change the training datasets (e.g., by including more data or omitting some data). For example, ASTRAEA [2] retrains SA models with generated text mutants to improve the fairness of SA systems. In-processing methods focus on improving the algorithms, e.g., by adopting adversarial debiasing [8] or adding discrimination-aware regularization terms [9] in models. Post-processing methods reduce bias by changing biased prediction results on the fly rather than retraining SA models. To the best of our knowledge, no post-processing methods for SA systems have been proposed in the literature.

Although pre-processing and in-processing methods have been successful in eliminating bias, both of them can be inapplicable under some circumstances. The two methods rely on retraining models, which can be time and resource intensive. Moreover, the pre-processing methods may require significant human efforts to collect and label more data. Furthermore, there is no guarantee that the retrained models maintain the same accuracy (as the original model) or is not biased in some other ways (a "fix" to a (fairness) "bug" may introduce another (fairness) "bug"). More importantly, pre-processing and in-processing methods require access to the SA engine and training data. This access is not available if one is using a third-party SA service. As not everyone is an SA expert or has sufficient data to train a good SA engine, many systems make use of such third-party SA service [10]. The above facts inspire us to propose and investigate several post-processing strategies that can help deal with SA unfairness.

In this paper, we propose a strategy to heal an SA engine biased output on the fly. This work is considering binary genders and binary sentiments. To uncover biased predictions, we make use of BiasRV, which adopts the *distributional fairness* concept [7]. Given an input text, BiasRV generates the same number of male and female mutants. Following distributional fairness, it is expected that the proportions of mutants predicted as positive for both genders (pos_F and pos_M) should be close enough. If $|pos_F - pos_M|$ is larger than a threshold, BiasRV regards the SA system as biased towards the input text. After biased predictions are detected, we propose an on-the-fly healing approach to construct fair predictions and force the biased SA system to return fair results. The constructed predictions should satisfy a property:

the SA system should return the same result for any text and its mutants. As a result, the constructed predictions will not violate distributional fairness.

Arguably, fairness is not the only goal for SA; accuracy is another goal (a fair SA engine will not be of use if it has low accuracy). Our goal is thus to have an on-the-fly healing solution that can produce fair results that can boost or at least not harm accuracy by much. As a preliminary exploration, this paper investigates 6 different self-healing strategies that can construct fair sentiment predictions when bias is detected (more details are given in Section III).

We evaluate the effectiveness of the 6 strategies on two datasets: IMDB [11] and SST [12]. On the IMDB dataset, the best performing strategy can return distributionally fair predictions and increase the accuracy by 2.76%. On the SST dataset, the best performing strategy can return distributionally fair predictions and improve the accuracy by 2.85%. The results highlight the feasibility of using the on-the-fly black-box healing method. It is worth mentioning that the self-healing method is only applied when biased predictions are detected, having no impact on other texts. Besides, our proposed method is a post-processing method, which can be used in conjunction with other bias elimination techniques (e.g., it can be applied to SA systems that are optimized with in-processing and post-processing methods) to improve their fairness guarantees further.

The rest of this paper is organized as follows. Section II introduces two preliminary concepts that are first proposed in BiasRV [7] and reused here: gender-discriminatory mutants and distributional fairness. Section III describes the six self-healing strategies we use to construct fair sentiment predictions. Section IV describes our experiments and results. We present related work in Section V. We conclude the paper and present future work in Section VI.

II. Preliminary

This section discusses some necessary preliminaries, including the gender-discriminatory mutant generation engine used in BiasFinder [1] to create templates for given texts and generate text mutants from the templates. It also describes the concept of distributional fairness used in BiasRV [7] to uncover biased predictions in SA systems at runtime.

A. Gender-Discriminatory Mutant Generation

The first step to uncover fairness violation is to generate gender-discriminatory mutants, which we define as the texts that only differ in those words reflecting gender information (e.g., names and pronouns) with the original text. BiasFinder [1] curates such mutants in two phases: (1) creating a template from a given text and (2) generating mutants by replacing placeholders in a template.

We view an input text I as a token sequence $(t_1, t_2, \cdots, t_n)$. BiasFinder leverages named entity recognition and coreference resolution to extract the *protected tokens*, which can divide a population into certain groups (e.g., male and female). In this paper, there are four types of protected tokens: names, subjective pronouns, possessive pronouns, and reflexive pronouns. In Figure 1, ⟨`Drew Barrymore`, `she`, `her`, `herself`⟩ are identified as protected tokens. Then BiasFinder substitutes protected tokens with placeholders to create a template.

Text
This is a real feel good film. **Drew Barrymore** is excellent again, **she** plays **her** part well and fulfills **herself**.

Generated Template
This is a real feel good film. ⟨**name**⟩ is excellent again, ⟨**subjective-pronoun**⟩ plays ⟨**possessive-pronoun**⟩ part well and fulfills ⟨**reflexive-pronoun**⟩.

Male Mutant
This is a real feel good film. **James** is excellent again, **he** plays **his** part well and fulfills **himeself**.

Female Mutant
This is a real feel good film. **Anne** is excellent again, **she** plays **her** part well and fulfills **herself**.

Fig. 1. An illustrative example of how BiasFinder generates gender-discriminatory mutants.

GenderComputer[1] provides a database of names from several countries. BiasFinder filters the names that are only used for one gender globally and divides them into two groups: male and female names. To generate a gender-discriminatory mutant given a template and a gender, BiasFinder first selects one name that belongs to that gender. After making the pronouns consistent with the gender, BiasFinder replaces placeholders in a template with selected names and pronouns to generate a mutant. For example, using the template in Figure 1, we can use ⟨`James`, `he`, `his`, `himself`⟩ to generate a male mutant and use ⟨`Anne`, `she`, `her`, `herself`⟩ to generate a female mutant. By default, BiasFinder generates 30 mutants for each gender if the protected tokens contain names.

B. Distributional Fairness in SA Systems

BiasRV [7] proposes the *distributional fairness* concept, which is used to analyze whether an SA system has different preferences for mutants of two genders generated from the same template. An SA system that satisfies distributional fairness should treat all the mutants matching the same template similarly. More specifically, for the two groups of text mutants, the distribution of predicted sentiments given by the SA system should be close enough.

Both sentiment and genders of mutants are set as binary in this work: two sentiments (positive and negative) and two genders (male and female). We create a template T from a piece of input text I, after which we generate a group of male mutants M and a group of female mutants F based on the template. Assuming that pos_F (pos_M) is the proportion of female (male) mutants predicted as positive (ranging from 0 to 1), we define *distributional fairness score* as $dfs = |pos_F - pos_M|$.

[1]https://github.com/tue-mdse/genderComputer

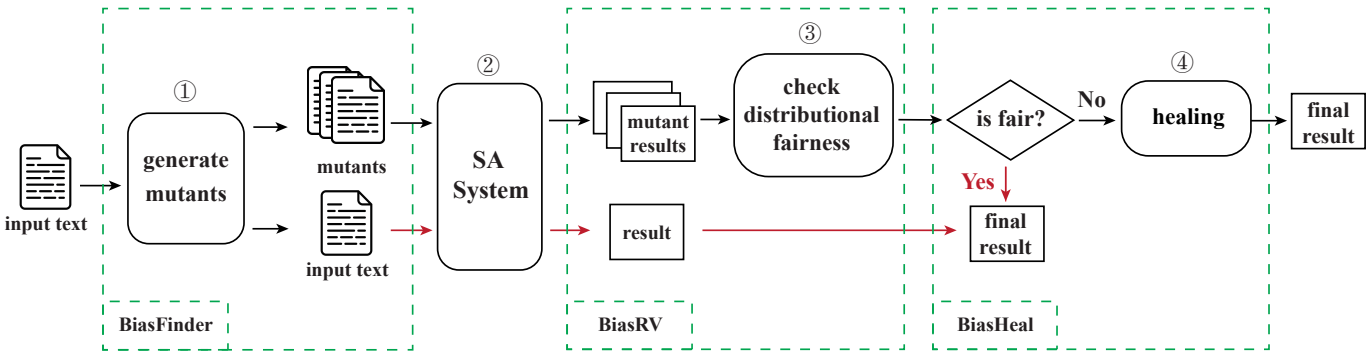

Fig. 2. The figure illustrates the workflow of bias detection and healing. A number of mutants are firstly generated based on the input text via BiasFinder, and then BiasRV computes a distribution fairness score for mutants to check the existence of bias in the SA system *with respect to* the input text. BiasHeal adopts a healing strategy to construct a fair final result if the SA system is biased. Otherwise, the original prediction for the input text is the final result.

For a given template, the *dfs* value reflects an SA system's degree of bias towards the genders. Larger *dfs* indicates that the SA system gives more different treatment to each gender; smaller *dfs* means that the SA system can treat male and female mutants derived from the same template more fairly. We set a *tolerance score* (α) that serves as a threshold to decide under what situation bias happens: if the distributional fairness score is higher than the tolerance score (i.e., $dfs \geq \alpha$), BiasRV flags that an SA system is biased for a template T. In this case, the SA system can potentially make biased predictions on instances (mutants) of the template.

III. Approach

This section describes our proposed approach. Our preliminary work is considering binary genders and binary sentiments. It first introduces an overview of our workflow from detecting bias to healing bias. Then, we discuss the post-processing bias healing method and propose six self-healing strategies considered in this preliminary work.

A. Overview

Fig 2 presents a high-level overview of how we make an SA system uncover and heal bias. In the figure, ② is a black-box SA service, which takes a piece of text as input and returns the predicted sentiment of the text as well as the corresponding confidence score for the result. When an SA system receives an input text, BiasFinder [1] first builds a template for the text and generates the same numbers of mutants in two genders (described in Section II-A)[2]. Then, the SA system predicts the sentiments of these mutants, and BiasRV [7] computes *dfs* (described in Section II-B) to analyze whether the SA system can potentially make biased predictions. If bias is detected, BiasHeal (④ in Fig 2) adopts a healing strategy to construct and return a fair result. We elaborate on the details of BiasHeal in Section III-B.

[2]We generate 30 mutants for each gender if there is a name in the text and generate 1 mutant for each gender if only pronouns are identified in the text.

B. On-the-Fly Bias Healing

Now we discuss BiasHeal, which performs as a post-processing method that works in a purely black-box manner to heal bias in a system that uses an SA service. When the SA service is detected to be potentially biased with respect to an input text, BiasHeal can construct a fair result to ensure the SA service generates predictions for mutants of the input text in a way that satisfies distributional fairness. We formalize the requirement as follows:

$$\forall I \text{ and } I', \text{if } t(I) = t(I'), \text{then } S(I) = S(I') \qquad (1)$$

where I and I' are two input texts. $t(\cdot)$ is a function to extract the template from a text, and $S(\cdot)$ is the output of the SA service given an input text. The above fairness specification proposes no requirement for the accuracy of the SA service. We hope that the healing strategy can improve accuracy and at the same time eliminate bias. However, past studies show that (in-processing) methods produce fairer results at the cost of accuracy [8].

In this paper, BiasHeal is adopted as a post-processing method and operates in a fully black-box manner without any access to SA engines or training datasets. The information that can be used for healing includes the extracted template from an input text, mutants generated from the template, and the prediction results from an SA system. In this preliminary work, we investigate 6 strategies; each of them **replaces the prediction of the input text with the following**:

1. **The prediction given to the majority of its mutants**: The fundamental hypothesis behind this strategy is that bias is occasional: the SA system can correctly predict the sentiments of most mutants. So we can count the occurrence of predictions given for all mutants and take the one that appears the most as the final result[3].
2. **The prediction given to the minority of its mutants**: The hypothesis behind this strategy is that bias is widespread among the mutants: the predicted results of mutants are mostly

[3]For Strategy 1 and 2, if the occurrences of the positive and negative sentiments in the predictions are equal, we break the tie by replacing the prediction of the input text with the positive sentiment.

biased.

3. **The prediction suggested by the average confidence score (ACS) of its mutants**: An SA service outputs the predicted result as well as a confidence value, which indicates to what extent the SA is confident that the input text is of positive sentiment[4]. However, such information is not considered in the previous two strategies. This strategy computes the average confidence score (ACS) for all the mutants. If the value is less than 0.5, this strategy outputs "negative" as the final sentiment (and "positive" otherwise).

4. **The prediction suggested by the ACS of its bottom 50% male mutants**: We find that male mutants tend to have a higher average confidence value than that of the female mutants (more details are given in Section IV-B1). If we compute the ACS for all male or female mutants, it emphasizes such bias. So this strategy computes the ACS of the bottom 50% male mutants and returns the corresponding result as a final prediction.

5. **The prediction suggested by the ACS of its top 50% female mutants**: For the same reason described in Strategy 4, we compute the ACS of the top 50% female mutants and return the corresponding result.

6. **The prediction suggested by the ACS of the mutants whose confidence scores fall in the interquartile range**: This strategy tries to discard the mutants with confidence scores that deviate much from the norm (i.e., median). We rank all the mutants according to their confidence values. Then, we return the result suggested by the ACS of the mutants whose confidence scores are in the middlemost 50% of the distribution.

IV. Research Questions and Evaluation

A. Experiment Setting

The transformer models have been shown to outperform the traditional SA tools [13]. We create two SA models by fine-tuning BERT [14] on two datasets containing movie reviews, which is suitable in our experiments since they indicate binary sentiments. One dataset is the IMDB dataset [11] that contains movie reviews from IMDB. Out of the original training set ($25,000$ texts), we extract $22,010$ texts without gender-related words (i.e., names and pronouns) as the training set. From all the texts in the dataset, we use the $6,532$ movie reviews that have gender-related words as the evaluation set. Another dataset is the Stanford Sentiment Treebank (SST) dataset, consisting of many single sentences extracted from movie reviews [12]. According to whether a text contains gender-related words, we split the SST dataset into $7,835$ texts for training and 339 texts for evaluation. The IMDB dataset contains longer texts (typically of a few sentences), while the SST dataset contains shorter texts. The replication package and datasets are made anonymous and available online[5].

[4]Note that if SA outputs a positive prediction, this confidence score is at least 0.5. On the other hand, if SA outputs a negative prediction, this confidence score is less than 0.5.

[5]https://anonymous.4open.science/r/BiasHeal_ICSME_NIER/

TABLE I
THE ACCURACY OF SA MODELS ON DIFFERENT DATASET.

Dataset	*Fair texts*		*Biased texts*	
	# of texts	accuracy	# of texts	accuracy
IMDB	6387	93.11%	145	66.21%
SST	304	89.14%	35	62.86%

TABLE II
THE ACCURACY OF DIFFERENT HEALING STRATEGIES.

	IMDB		SST	
	accuracy	difference	accuracy	difference
Strategy 1	**68.97%**	**+2.76%**	54.29%	−8.57%
Strategy 2	31.03%	−35.18%	45.71%	−17.15%
Strategy 3	64.83%	−1.38%	60.00%	−2.86%
Strategy 4	57.93%	−8.28%	60.00%	−2.86%
Strategy 5	64.83%	−1.38%	**65.71%**	**+2.85%**
Strategy 6	64.14%	−2.07%	60.00%	−2.86%
No Strategy	66.21%		62.86%	

B. Research Questions

Here, we introduce the research questions to be explored in this work, and our experimental results that answer them.

1) RQ1. Do the SA models perform differently when bias happens?: For an input text with a gender-related word, we first decide whether an SA system has bias with respect to the input text. We divide input texts into two groups: 'fair texts' and 'biased texts'. Then, we evaluate the model accuracy on the two groups of texts.

Table I shows the results, respectively. We find that for both datasets, SA systems have high accuracy on *fair texts*. The accuracies on *biased texts* drop significantly, by 26.90% and 26.28% for IMDB and SST datasets respectively. It means that compared with the performance on other inputs, SA models perform much worse on texts towards which they have bias. We analyze the confidence scores of mutants further. We find that, in general cases, the SA systems assign higher average confidence scores to male mutants. It indicates that the distributional fairness concept does capture gender bias in SA systems. More specifically, both the two SA systems evaluated in this paper give more preference to positive sentiment when predicting male mutants. This also inspires Strategy 4 and 5 in Section III-B, which aims to shift such biased preference.

2) RQ2. Which healing strategy can achieve the highest accuracy?: In this RQ, we apply the 6 healing strategies mentioned in Section III-B and measure the accuracy of the fair predictions produced by each strategy. We compare their results with the accuracy that the SA models (with no fairness healing strategy) can achieve on 'fair texts' reported in our answer to in RQ1. This is done to analyze the impact of enforcing fairness on the fly, aiming to find the healing strategy that can achieve the highest accuracy.

Table II shows the results. We find that on the IMDB dataset, Strategy 1 can achieve the highest accuracy (68.97%, corresponding to an accuracy increase of 2.76% compared to the accuracy reported in RQ1). On the SST dataset, Strategy 5 can achieve higher accuracy than when no fairness healing

is performed (by $+2.85\%$). However, for this dataset, Strategy 1 performs worse than when no healing is performed (by -8.57%). Note that the lengths of texts in the IMDB dataset are longer. Such texts are likely to be more robust to the modification in gender-related words. As a result, an SA system can predict most mutants correctly, which satisfies the hypothesis behind Strategy 1. The texts in the SST dataset are shorter and more sensitive to the modification, so both Strategy 1 and 2 are underperformed (as compared to the other strategies that are based on confidence). The evaluation results suggest that different strategies may perform better for datasets with different properties, e.g., text lengths. We leave further investigation into this in future work.

V. Related Work

In the software engineering literature (and beyond), researchers have recently been interested in building fair software. These studies mainly focus on (AI) software that takes tabular data (e.g., Adult Census Income dataset[6]) as input [4], [15], [16], [17]. However, a few (like this work) have focused on (AI) software that process natural language text [1], [2], [5], [6], [7]. We have highlighted these closely related works in Section I. Here we discuss several commonly used fairness concepts.

Individual fairness requires that similar individuals should receive similar outcomes. The metamorphic relationship used by [1], [2], [5], [6] (i.e., an SA system should have the same prediction for a pair of texts that only differ in gender-related words) is an instance of individual fairness. Group fairness is the goal that based privileged and unprivileged groups will be treated similarly. Researchers have proposed various group fairness metrics to quantify the bias in models. For example, Fairway [16] computes Equal Opportunity Difference (EOD) and Average Odds Difference (AOD) [18] to measure fairness. However, computing these metrics requires a large population, and they can only measure an algorithm's *overall* fairness. This paper uses the distributional fairness concept [7] to decide whether an SA system makes biased predictions on a *specific* input.

VI. Conclusion and Future Work

In this paper, we propose a black-box post-processing method to make an SA system heal bias and construct fair results when bias is detected. For an input text, we use BiasFinder to extract protected tokens to build a template and generate mutants of two genders. Then, we use BiasRV to check whether the prediction results of these mutants satisfy distributional fairness. If biased is detected, we adopt a healing strategy to construct a distributionally fair prediction. We propose and investigate 6 different healing strategies on the IMDB and SST datasets. The evaluation results show the SA systems perform worse when bias happens, and we can leverage extra information of predicted results for mutants to improve accuracy at the same time as healing bias. Our results also show that the best-performing strategy can vary on datasets with different properties, e.g., text lengths, suggesting the need to carefully choose a suitable healing strategy for a specific dataset.

In the future, we plan to validate our findings on more SA systems and datasets. We also plan to design additional post-processing healing methods and consider the effectiveness of combining multiple (pre-processing, in-processing, or post-processing) healing methods together.

[6]https://archive.ics.uci.edu/ml/datasets/adult

References

[1] M. H. Asyrofi, I. N. B. Yusuf, H. J. Kang, F. Thung, Z. Yang, and D. Lo, "Biasfinder: Metamorphic test generation to uncover bias for sentiment analysis systems," *CoRR*, vol. abs/2102.01859, 2021.

[2] E. O. Soremekun, S. Udeshi, and S. Chattopadhyay, "Astraea: Grammar-based fairness testing," *CoRR*, vol. abs/2010.02542, 2020.

[3] High-Level Expert Group on AI, "Ethics guidelines for trustworthy ai," European Commission, Brussels, Report, Apr. 2019. [Online]. Available: https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai

[4] R. Angell, B. Johnson, Y. Brun, and A. Meliou, "Themis: Automatically testing software for discrimination," in *ESEC/FSE 2018*. New York, NY, USA: ACM, 2018, p. 871–875.

[5] S. Kiritchenko and S. Mohammad, "Examining gender and race bias in two hundred sentiment analysis systems," in *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*. New Orleans, Louisiana: ACL, Jun. 2018, pp. 43–53.

[6] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, "Beyond accuracy: Behavioral testing of NLP models with CheckList," in *ACL*. Online: ACL, Jul. 2020, pp. 4902–4912.

[7] Z. Yang, M. Hilmi Asyrofi, and D. Lo, "BiasRV: Uncovering Biased Sentiment Predictions at Runtime," *arXiv e-prints*, p. arXiv:2105.14874, May 2021.

[8] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. New York, NY, USA: ACM, 2018, p. 335–340.

[9] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Fairness-aware classifier with prejudice remover regularizer," in *ECML-PKDD*. Berlin, Heidelberg: Springer-Verlag, 2012, p. 35–50.

[10] D. Georgiou, A. MacFarlane, and T. Russell-Rose, "Extracting sentiment from healthcare survey data: An evaluation of sentiment analysis tools," in *2015 Science and Information Conference (SAI)*, 2015, pp. 352–361.

[11] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis." Portland, Oregon, USA: ACL, June 2011, pp. 142–150.

[12] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. USA: ACL, 2005, p. 115–124.

[13] T. Zhang, B. Xu, F. Thung, S. A. Haryono, D. Lo, and L. Jiang, "Sentiment analysis for software engineering: How far can pre-trained transformer models go?" in *ICSME*, 2020, pp. 70–80.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *ACL*. Minneapolis, Minnesota: ACL, Jun. 2019, pp. 4171–4186.

[15] J. M. Zhang and M. Harman, ""ignorance and prejudice" in software fairness," in *ICSE*, 2021, pp. 1436–1447.

[16] J. Chakraborty, S. Majumder, Z. Yu, and T. Menzies, "Fairway: A way to build fair ml software," in *ESEC/FSE*. New York, NY, USA: Association for Computing Machinery, 2020, p. 654–665.

[17] P. Zhang, J. Wang, J. Sun, G. Dong, X. Wang, X. Wang, J. S. Dong, and T. Dai, "White-box fairness testing through adversarial sampling," in *ICSE*, 2020, pp. 949–960.

[18] R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, S. Nagar, K. N. Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang, "Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias," 2018.