

# Paper Review

Benjamín Farías Riquelme<sup>1</sup>

Universidad de Chile  
`benjamin.farias@ing.uchile.cl`

**Abstract.** [...]

**Keywords:** Natural Language Processing · Counterfactual Explanation · NLP interpretability

## 1 Introduction

Alongside the increase in capacity and influence of algorithms, there is an increase in the concerns and risks of them inadvertently perpetuating human biases [7,26]. This phenomenon is particularly evident in the context of neural networks developed for Natural Language Processing (NLP), as they learn directly from human-generated texts. The delegation of the decision-making process to these algorithms has the potential to engender negative societal impact if there are undetected or neglected biases [4,11,16,23,35]. We will use the term predictive model (in NLP), to refer to any model that takes text as input, and produce a prediction, decision or classification as output (e.g. toxic text classification or spam filtering). To illustrate the risks of unassessed biases in predictive models in NLP, consider the following toy example.

A certain company is accused of favoring men over women in their hiring process. To solve this problem, the company decides to leave the process of a neural network, which determines if an applicant should be hired or not, based on their anonymized resume. The network is trained with the data of the previous processes, so it replicates the same biases. The next time the company is questioned for their hiring process, they respond that it is managed by an algorithm and, unlike humans, algorithms are objective, so if more men than women are hired, it must be because there are more men better qualified for the job.

In the example, the bias is the association of gender with qualification for the job, and it is originated by the use of biased training data. In Section 2 we provide a more precise definition of bias in NLP and its origins.

A substantial corpus of research has been dedicated to the examination of prediction bias in NLP, and biases in NLP in general. Most of them can be encapsulated in three categories: characterization of bias and its risks, measuring bias, and debiasing. The last two categories are closely intertwined, given that

debiasing methods, which aim to remove a specific bias, often do so by reducing the metrics defined in the bias measuring literature. However, it has been observed that the metric reduction approach is more likely an elimination of symptoms rather than biases [2,12,13].

In general, bias measuring works choose a social bias, such as gender or racial bias, and propose a metric to quantify the presence of this bias in either the model representations or responses. These are referred in the literature as intrinsic and extrinsic metrics, respectively [8,16]. This concept is closely related to that of fairness metrics [7], which, in turn, are metric directly designed to measure the consequences and symptoms of biases (or other problematic elements), rather than the bias itself.

We propose an alternative approach to addressing the issue of bias in predictive models. Rather than measuring the extent of bias in a model, we endeavor to identify the biased associations that the model makes. To better explain this idea, we will continue with the hiring model example.

An association against gender bias, determined to demonstrate that the model used by the company is biased, collects 100 applicants and their results. They find that men were accepted 4 times more than women, even when they have similar background. With this fairness test, the association is sure that the model favors men over women, however, the company points out a key detail, the resumes are anonymized, so the model does not have any information regarding gender, and thus cannot be biased. The association reviews the applications again, this time focusing on the difference between accepted and rejected resumes. They find that the rejected resumes tend to use longer sentences with more unique words, and that this characteristic is more frequent in women's resumes<sup>1</sup>. After editing the resumes to have a similar style and passing them to the model, it is found that the rejection rate is now equitable between men and women, proving that it was, in fact, biased.

Here, even if a metric indicates that the answers of the model are biased, it does not explain why it happens. After examining a set of input-output samples, it is found a correlation between a single attribute of the input and a particular output, an association can be called a bias. We denote this procedure of finding associations as bias detection. In the example, the association is identified through manual examination. However, it would be preferable to have a mechanism capable of automatically detecting bias.

In this short survey we review some methods that can be employed, or repurposed, to perform this task. The scope of the survey is restricted to methods that can be applied on predictive models with transformer architecture. We divide the methods in two categories: examination methods (Section 4.2), that examines the representations used or generated by the model and its operations, and example generation methods (Section 5), that generates examples that might show the biases of the model. We also indicate if the methods are

---

<sup>1</sup> This toy example is based on the findings of Qu et al.[29]

useful for a confirmatory or exploratory bias analysis. The definitions of both analyses are provided in Section 3.

	Examination	Example Generation	Confirmatory	Exploratory
WEAT	✓		✓	
SEAT	✓		✓	
CEAT	✓		✓	
RIPA	✓		✓	
AG	✓			✓
MiCE		✓		✓
GYC		✓		✓
POLYJUICE		✓		
PIS		✓		✓
MEIO		✓		✓
R-LACE	✓		✓	✓
TEA	✓		✓	✓

Table 1. Caption.

## 2 Bias in Predictive Model for NLP

Despite of the large number of works addressing bias in NLP, there is a lack of consensus regarding the definition of bias [4]. The discussion over the different definitions of bias constitutes a complex subject that will not be thoroughly addressed in this survey. By prediction bias in NLP, we refer to the prior that informs a predictive model to make its predictions [2,35]. In essence, prediction biases are associations between features of the input and the output, encoded in the model. Following this definition, all models have biases, and it is not something inherently problematic, but another gear in the model’s mechanism.

Biases can be harmful when they come from harmful precedents [6]. Biases that are not aligned with reality, or are aligned with a reality that we do not wish the model to learn from, are denominated unintended biases [2,35]. This would be the case the association between qualification for the job and a feature that correlate with gender in the hiring model example.

The majority of predictive models in NLP are trained with real-world text samples, which are unavoidably biased by the context in which they are written and the demographic of who writes them [9,17,28]. In consequence, there is a high chance that the models replicate those biases. This can lead models to pick up patterns that do not generalize to other contexts or demographics, or rely on undesired relations, resulting in unfair or harmful predictions [1,23,26,35]. Even if the data does not present undesired biases, models themselves can display unintended biased behavior due to certain design choices [23], or inherit them from biased representations [5,6].

Unintended biases, for prediction tasks in NLP, can be divided into four categories according to the source of the bias [35]:

- **Label Bias:** Emerges when the model learns predictions that diverge substantially from the ideal distribution, product of labels aligned with a (not desired) biased reality.
- **Selection Bias:** Emerges when the model learns from data that is non-representative of the distribution to where it would be applied.
- **Overamplification:** Emerges when the model itself pick up small difference in the data, and amplify them to be much larger in the predicted outcomes.
- **Semantic Bias:** Emerges when the embeddings used by the model encode biased relations.

### 3 Indirect Effect

In our hiring model example (Section 1), the model express a gender bias, but the actual association encoded in the model is between the length of sentences and variety of words, which correlates with gender, and the acceptability for the job. In fact, the model does not even recieve any information regardign gender. In this toy scenario, gender has no direct effect, but it has a significant indirect effect.

The effect that a variable has over an outcome can be decomposed as the sum of its direct effect and its indirect effect. The direct effect is a quantification of the influence that a variable has on an outcome, that is not mediated by other variables. Let  $X$  be the varieble whose effect we seek to assess,  $Y$  the response variable, and  $Z$  the set of all the intermediate variables between  $X$  and  $Y$ . The direct effect of  $X$  over  $Y$  measures the sesivity of  $Y$  to changes in  $X$ , while  $Z$  is held fixed [27]. Formaly, the direct effect of an event  $X = x$  is given by

$$DE(x, x^*; Y) = Y_{xZ_x} - Y_{x^*} \quad (1)$$

where  $x^*$  is a reference value for  $X$  and the notation  $Y_x$  is used to represent the value  $Y$  would attain when  $X$  is set to be  $x$ . In contrast, the indirect effect of an event  $X = x$  quantify the sensitivity of  $Y$ , to changes in the mediators  $Z$  induced by  $X = x$ . This is equivalent to measure the change in  $Y$  when  $Z$  is set to the value it would attain under  $X = x$ , while  $X$  is held fixed at the reference value [27]. The indirect effect is given by

$$IE(x, x^*; Y) = Y_{x^*Z_x} - Y_{x^*} \quad (2)$$

The indirect effect is an important factor to consider when analysing bias that is often neglected. In the hiring model example, the idea that the model a bias realated to  $X$  (gender) is rejected at first, because is obvious that  $X$  has no direct on  $Y$  (the output). The iderect effect, and subsequently the bias, is probed after finding, by chance, the mediator  $Z$  (length of sentences and variety of words) between  $X$  and  $Y$ . In this case is evident that  $X$  has no direct effect,

as it is not part of the input, but there can be similar scenarios where  $X$  is in the input. Consider the following alternative version of the example:

The company, aware that their model will be accused of perpetuating gender bias, include the gender of the applicants, adding two identical copies of each of them, varying only the gender, to the training data, in order to prevent the bias. After putting the model in use, the company is denounced by the association, alleging that the model perpetuates gender bias.

In this alternative version of the example,  $X$  is modified in the input, but  $Z$  remains in its original value, so the model learns the same association between  $Z$  and  $Y$ . The direct effect of  $X$  on  $Y$  is low, but the bias persists. In order to achieve a complete bias detection, it is imperative to ensure that the detection method is assessing the total effect and not only the direct effect.

If, as in the example, bias is dominated by an indirect effect, it may be necessary a characterization of the mediator variables  $Z$  to effectively debias the model. However, in general,  $Z$  is an unknown variable, and can correspond to a feature that is not explicitly represented in the data. While  $X$  can be directly observed and intervened,  $Z$  must be found. We denote as confirmatory analysis to verify or measure the effect of  $X$  over  $Y$ , and as exploratory analysis to verify or measure the effect of a mediator  $Z$ , between  $X$  and  $Y$ . In other words, confirmatory detection methods detect an association that involves a target variable, and exploratory detection methods can detect an association that involves a previously unknown variable.

## 4 Mechanism Examination

### 4.1 Representations

This category encompasses methods that can be employed to detect biases in the word representations used by the model. These can be either the word-embeddings used in the input, or the contextualized embeddings generated by the model.

**Word-Embedding Association Test** The Word-Embedding Association Test (WEAT) [6] is one of the most influential work addressing bias encoded in word-embeddings. WEAT is an adaptation of the Intrinsic-Association Test used in social psychology, to measure stereotype-related bias in word-embeddings. Given 2 sets  $X, Y$  of target words (e.g. professions) and 2 sets  $A, B$  of attributes words (e.g. gender nouns), WEAT provides a metric (equations 3 and 4) that measures the differential association of the two sets of target words  $X, Y$  with the attributes  $A, B$ , where the association between words is defined as the cosine similarity.

$$\text{WEAT}(A, B, X, Y) = \frac{\text{mean}_{x \in X} s(x, A, B) - \text{mean}_{y \in Y} s(y, A, B)}{\text{sd}_{w \in X \cup Y} s(w, A, B)} \quad (3)$$

$$s(w, A, B) = \text{mean}_{a \in A} \text{cossim}(w, a) - \text{mean}_{b \in B} \text{cossim}(w, b) \quad (4)$$

WEAT was designed to be applied in contexts where the words of  $X$  and  $Y$  should be equally associated to the words of both  $A$  and  $B$ . If, for example, it is found  $A$  is more associated with  $X$  than  $Y$ , it said that the embeddings are biased.

Many works have adapted WEAT to work in other scenarios. For instance, SEAT [22] measures association in sentences, replacing the word-embeddings by sentence-embeddings, and CEAT [15] measures association in contextualized embedding, by computing WEAT  $N$  times with random contextualized embeddings, from different sentences containing words from the target and attribute sets, and then analizing the resulting distribution. There are also alternative formulations of WEAT for the same context, such as RIPA [10], that propose to use the inner product instead of cosine similarity to measure association.

WEAT-based methods can be employed for detecting bias, by defining a threshold for the bias metric delimit from which point the embeddings are considered to be biased. However these methods are constricted by the requirement of defining the target and attribute sets. WEAT can only look for pre-determined associations, and is susceptible to error if word sets are not well defined.

A possible solution to the limitation of the target or attribute sets, could be to develop an algorithm that automatically generates these sets, similar to how the Intersectional Bias Detection method iterate [15], proposed by the authors of CEAT, iterate over different combinations of subsets of the attributes to find biases associated to individuals that are in the intersection of two groups.

**Analogy Generation** One interesting feature of word-embeddings is that they have been found able to express words relation through vector difference [24,34]. For example, the difference between the embeddings for man and woman is similar to the difference between the embeddings for king and queen. This can be expressed in an analogy of the form “man is woman as king is to queen”. Given a pair of words  $x$  and  $y$ , the method of analogy generation [5] consist in looking for pairs  $(a, b)$  that might fit in the analogy “ $x$  is to  $y$  as  $a$  is to  $b$ ”. To do this, each pair  $(a, b)$  is assigned a score defined by:

$$S_{x,y}(a, b) = \begin{cases} \text{cossim}(x - y, a - b) & \text{if } \|a - b\| \leq \delta \\ 0 & \text{if } \|a - b\| > \delta \end{cases} \quad (5)$$

where  $\delta$  is a threshold for the distance between  $a$  and  $y$ . Analogy generation can be employed for bias detection at the level of the word-embeddings, but have similar issues to the ones of WEAT, as it requires to pre-define a set of candidate words for  $a$  and  $b$ , which can be under-representative.

**R-LACE** Relaxed Linear Adversarial Concept Erasure (R-LACE) [31] is method designed for the task of concept erasure in the representations. That is, given a

set of vector representations  $X = \{x_i\}_{i=1}^N \subseteq \mathbb{R}^d$  (e.g. word-embeddings) and a set of response variables  $Y = \{y_i\}_{i=1}^N$  that indicates a concept in the vectors (e.g. gender), implement some function  $r : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ , such that the resulting vectors  $r(x_i)$  preserve as much information as possible, while not being predictive of concept  $Y$ .

To erase a concept, R-LACE finds a subspace  $B \subseteq \mathbb{R}^d$  that contains the information of the target concept, within the representations, and project the vector representations to the orthogonal complement of  $B$ . The subspace  $B$  is determined by solving the minmax problem:

$$\min_{\theta} \max_P \sum_{i=1}^N \mathcal{L}(y_i, g^{-1}(\theta^T P x_i)) \quad (6)$$

where  $f_{\theta}(x) = g^{-1}(\theta^T x)$  is a generalized linear model, with parameters  $\theta$  and link function  $g$ ,  $\mathcal{L}$  is a loss function, and  $P$  is a  $d \times d$  orthogonal projection matrix that neutralizes a rank  $k$  subspace, with  $k$  being an hyper-parameter of the algorithm.

Note that the definition of R-LACE does not require to have an explicit definition of the concept to be erased, just to know the response variables. R-LACE can be repurposed for bias detection, at the level of the inner representations, by erasing the concept that determines a particular prediction. R-LACE can be expanded for non-linear subspaces by applying a kernel on  $f_{\theta}$  [32].

Let  $X$  be the inner representations given by some inner layer of a predictive model,  $Y$  a response variable that indicates if a representation  $x_i$  is assigned to particular prediction or not by the model, and  $r(X) = \{r(x_i)\}_{i=1}^N$  the resulting vectors after applying R-LACE on  $X$  to erase  $Y$ . If there are two instances  $a$  and  $b$ , such that their representations are similar after applying R-LACE,  $r(x_a) \approx r(x_b)$ , but not before,  $x_a \not\approx x_b$ , then the difference between  $a$  and  $b$  might show the prediction bias.

## 4.2 Model Mechanism

**Targeted Edge Ablation** Targeted Edge Ablation (TEA) [18] is a technique designed to remove an specific behavior of the model, by ablating a small number of edges, or pathways, between its components. In this context, given a model  $M$  and a loss function  $\mathcal{L}$  (that is not necessarily the one used to train  $M$ ), a behavior is specified as a set of inputs  $\mathcal{D}$  on which  $M$  achieves low loss. The task of behavior removal is defined as modifying  $M$  to create another model  $M'$  that achieves high loss on  $\mathcal{D}$ , without a significant increase of loss on inputs outside  $\mathcal{D}$ .

To ablate  $M$ , first is necessary to choose at what level of granularity represent the model's computations, and write the graph  $G$  that describes  $M$  at that specific level (e.g. represent  $M$  as a graph of attention heads and feed forward layers). Then the ablated edges in  $G$  are determined by solving:

$$\min_W \mathcal{L}(G_W, D_{\text{train}}) - \alpha \mathcal{L}(G_W, \mathcal{D}) + \lambda(t) R(W) \quad (7)$$

where  $G_W$  is  $M$  with a mask  $W$  applied over the edges of  $G$ ,  $D_{\text{train}}$  is a set of train data, disjoint to  $\mathcal{D}$ ,  $R$  is a regularization function,  $\alpha$  is a constant, and  $\lambda(t)$  a regularization weight that increase over time. The mask  $W$  assigns to each edge  $e = (A, B)$  of  $G$  a weight  $w_e \in [0, 1]$ , such that node  $B$  receives the following combination of the original value  $v_A$  and the ablated value  $\mu_A$  from node  $A$ :

$$w_e v_A + (1 - w_e) \mu_A \quad (8)$$

After the optimization is finished, the edges whose weight does not surpass a specified threshold are ablated.

If  $\mathcal{D}$  is set to represent a biased behavior, and TEA is capable of effectively removing it from the model, then that would confirm that  $M$  computes the specified bias. Moreover, if combined with a method to analize the representations, it could help to identify what is the bias association being computed by  $M'$ , by comparing the inner representations before and after the ablation.[21]

## 5 Example Generation

### 5.1 Counterfactual Examples

In machine learning, a counterfactual explanation is an example that illustrates how a different input would result in a different output [20]. In NLP task, such as text classification or next token prediction, for instance, if an input text  $X$  gets an output  $y$  by the model, a counterfactual example would be any text  $X'$ , similar to  $X$ , that yields a different output  $y'$  [14,37]. Though this definition is clear and widely used, it may be too narrow, as it leaves out the sense probability in the predictions.

**POLYJUICE** The POLYJUICE [38] method employs a fine-tuned GPT-2 [30] model to generate counterfactual examples. The model receives an input text  $X$  and a perturbation instruction, such as negation, insertion or deletion, and returns a modified version of  $X$ , following the given instruction.

**MiCE** The Minimal Contrastive Editing (MiCE) [33] method generates counterfactual examples via a masked language model, called editor model, that is trained to fill the masked spaces in an input text with tokens, such that the resulting text would obtain a given prediction by some predictive model  $M$ . The editor model receives the masked text and the target label as inputs. During training, the top  $n_1\%$  tokens with highest gradient attribution [36], towards the target prediction, are masked. To generate the counterfactuals, the percentage of masked tokens is varied between 0% and 55%, using binary search to find the optimal percentage and beam search to keep track of the edits. The generation process stops once an edit changes the prediction to the target.

**GYC** The GYC [19] method generates  $k$  counterfactual examples, for a given input text  $X$  and a condition  $C$ , through controlled text generation, without requiring to train or fine-tune a model. The method consists in modeling the distribution  $p(\tilde{y}|X, C)$ , where the condition  $C$  can be any restriction over the text, such as a class label.

Let  $LM$  be a language model transformer;  $LM$  generates a token  $y_t$  conditioned on the past tokens  $y_{<t} = \{y_i\}_{i=0}^{t-1}$  as follows:

$$\begin{aligned} o_t, H_t &= LM(y_{t-1}, H_{t-1}) \\ y_t &\sim \text{Categorical}(o_t) \end{aligned} \tag{9}$$

where  $H_{t-1}$  is the history matrix that captures the dependency of  $y_{t-1}$  on past tokens and  $o_t$  are the logits to sample  $y_t$  from a categorical distribution. To generate text conditioned on  $C$ ,  $H$  is perturbed two times, first to create  $\tilde{H}_t$  which enforces the reconstruction of  $X$ , and next to create  $\hat{H}_t$  which enforces the condition. To learn the perturbations a linear combination of three loss functions is employed, one for reconstruction and one enforce the condition, plus another one to ensure diversity. The reconstruction loss maximizes the log probability of the input text, the condition loss maximizes a score associated to the condition and the diversity loss maximizes the entropy of the generated logits.

## 5.2 Activation Maximization

**Preferred Input Synthesis** The Preferred Input Synthesis (PIS) [25] method generates inputs that maximize the activation of a target neuron (including the output logits), via optimization at the level of the latent space of a generative model. Given a predictive model  $M : X \rightarrow Y$  and a generative model  $G : Z \rightarrow X$ , denoting by  $M_h$  the activation of the target neuron  $h$ , PIS generate samples by solving:

$$\arg \max_{z \in Z} M_h(G(z)) - \lambda \|z\| \tag{10}$$

where  $\lambda$  is a regularization term and the optimization is performed with gradient descend. PIS was originally formulated for image generation, but can easily be re-adapted for text.

**Momentum Evolutionary Input Optimization** The Evolutionary Input Optimization (MEIO) [3] method is a generic framework for Activation Maximization that propose a model-agnostic approach, with a zero-order optimization on the latent space of a generative model. Given a predictive model  $M : X \rightarrow Y$ , a generative model  $G : Z \rightarrow X$ , and a target prediction  $y \in Y$ , MEIO generate samples by solving:

$$\min_{z \in Z} \mathcal{L}(M(G(z)), y) \tag{11}$$

where  $\mathcal{L}$  is a loss function. The optimization is performed via an evolutionary strategy with momentum updates, that involves iteratively updating a set of candidate solutions, by adding to them an independently sampled gaussian noise, which is combined with noised added in the previous iteration (the momentum).

## 6 Discussion

### References

1. Angwin, J., Larson, J., Mattu, S., Kirchner, L.: Machine bias. In: Ethics of data and analytics, pp. 254–264. Auerbach Publications (2022)
2. Bansal, R.: A survey on bias and fairness in natural language processing (2022), <https://arxiv.org/abs/2204.09591>
3. Barbalau, A., Cosma, A., Ionescu, R.T., Popescu, M.: A generic and model-agnostic exemplar synthetization framework for explainable ai. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part II. pp. 190–205. Springer (2021)
4. Blodgett, S.L., Barocas, S., Daumé III, H., Wallach, H.: Language (technology) is power: A critical survey of “bias” in NLP. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 5454–5476. Association for Computational Linguistics, Online (Jul 2020). <https://doi.org/10.18653/v1/2020.acl-main.485>, [https://aclanthology.org/2020.acl-main.485/](https://aclanthology.org/2020.acl-main.485)
5. Bolukbasi, T., Chang, K.W., Zou, J.Y., Saligrama, V., Kalai, A.T.: Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 29. Curran Associates, Inc. (2016), [https://proceedings.neurips.cc/paper\\_files/paper/2016/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2016/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf)
6. Caliskan, A., Bryson, J.J., Narayanan, A.: Semantics derived automatically from language corpora contain human-like biases. Science **356**(6334), 183–186 (2017). <https://doi.org/10.1126/science.aal4230>, <https://www.science.org/doi/abs/10.1126/science.aal4230>
7. Corbett-Davies, S., Gaebler, J.D., Nilforoshan, H., Shroff, R., Goel, S.: The measure and mismeasure of fairness. J. Mach. Learn. Res. **24**(1) (Mar 2024)
8. Czarnowska, P., Vyas, Y., Shah, K.: Quantifying social biases in nlp: A generalization and empirical comparison of extrinsic fairness metrics. Transactions of the Association for Computational Linguistics **9**, 1249–1267 (11 2021). [https://doi.org/10.1162/tacl\\_a\\_00425](https://doi.org/10.1162/tacl_a_00425), [https://doi.org/10.1162/tacl\\_a\\_00425](https://doi.org/10.1162/tacl_a_00425)
9. Eisenstein, J., O’Connor, B., Smith, N.A., Xing, E.: A latent variable model for geographic lexical variation. In: Proceedings of the 2010 conference on empirical methods in natural language processing. pp. 1277–1287 (2010)
10. Ethayarajh, K., Duvenaud, D., Hirst, G.: Understanding undesirable word embedding associations (2019), <https://arxiv.org/abs/1908.06361>
11. Field, A., Coston, A., Gandhi, N., Chouldechova, A., Putnam-Hornstein, E., Steier, D., Tsvetkov, Y.: Examining risks of racial biases in nlp tools for child protective services. In: Proceedings of the 2023 ACM Conference on Fairness, Accountability,

- and Transparency. p. 1479–1492. FAccT '23, Association for Computing Machinery, New York, NY, USA (2023). <https://doi.org/10.1145/3593013.3594094>, <https://doi.org/10.1145/3593013.3594094>
12. Goldfarb-Tarrant, S., Marchant, R., Muñoz Sánchez, R., Pandya, M., Lopez, A.: Intrinsic bias metrics do not correlate with application bias. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) 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). pp. 1926–1940. Association for Computational Linguistics, Online (Aug 2021). <https://doi.org/10.18653/v1/2021.acl-long.150>, <https://aclanthology.org/2021.acl-long.150/>
  13. Gonen, H., Goldberg, Y.: Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them (2019), <https://arxiv.org/abs/1903.03862>
  14. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking. Data Mining and Knowledge Discovery pp. 1–55 (2022)
  15. Guo, W., Caliskan, A.: Detecting emergent intersectional biases: Contextualized word embeddings contain a distribution of human-like biases. In: Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society. p. 122–133. AIES '21, Association for Computing Machinery, New York, NY, USA (2021). <https://doi.org/10.1145/3461702.3462536>, <https://doi.org/10.1145/3461702.3462536>
  16. Gupta, V., Narayanan Venkit, P., Wilson, S., Passonneau, R.: Sociodemographic bias in language models: A survey and forward path. In: Faleńska, A., Basta, C., Costa-jussà, M., Goldfarb-Tarrant, S., Nozza, D. (eds.) Proceedings of the 5th Workshop on Gender Bias in Natural Language Processing (GeBNLP). pp. 295–322. Association for Computational Linguistics, Bangkok, Thailand (Aug 2024). <https://doi.org/10.18653/v1/2024.gebnlp-1.19>, [https://aclanthology.org/2024.gebnlp-1.19/](https://aclanthology.org/2024.gebnlp-1.19)
  17. Kern, M.L., Park, G., Eichstaedt, J.C., Schwartz, H.A., Sap, M., Smith, L.K., Ungar, L.H.: Gaining insights from social media language: Methodologies and challenges. Psychological methods **21**(4), 507 (2016)
  18. Li, M., Davies, X., Nadeau, M.: Circuit breaking: Removing model behaviors with targeted ablation (2024), <https://arxiv.org/abs/2309.05973>
  19. Madaan, N., Padhi, I., Panwar, N., Saha, D.: Generate your counterfactuals: Towards controlled counterfactual generation for text. Proceedings of the AAAI Conference on Artificial Intelligence **35**(15), 13516–13524 (May 2021). <https://doi.org/10.1609/aaai.v35i15.17594>, <https://ojs.aaai.org/index.php/AAAI/article/view/17594>
  20. Madsen, A., Reddy, S., Chandar, S.: Post-hoc interpretability for neural nlp: A survey. ACM Comput. Surv. **55**(8) (dec 2022). <https://doi.org/10.1145/3546577>, <https://doi.org/10.1145/3546577>
  21. Marks, S., Tegmark, M.: The geometry of truth: Emergent linear structure in large language model representations of true/false datasets (2024), <https://arxiv.org/abs/2310.06824>
  22. May, C., Wang, A., Bordia, S., Bowman, S.R., Rudinger, R.: On measuring social biases in sentence encoders (2019), <https://arxiv.org/abs/1903.10561>
  23. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. ACM Comput. Surv. **54**(6) (Jul 2021). <https://doi.org/10.1145/3457607>, <https://doi.org/10.1145/3457607>
  24. Mikolov, T., Yih, W.t., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 conference of the north american chap-

- ter of the association for computational linguistics: Human language technologies. pp. 746–751 (2013)
25. Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., Clune, J.: Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 29. Curran Associates, Inc. (2016), [https://proceedings.neurips.cc/paper\\_files/paper/2016/file/5d79099fcdf499f12b79770834c0164a-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2016/file/5d79099fcdf499f12b79770834c0164a-Paper.pdf)
  26. O’neil, C.: Weapons of math destruction: How big data increases inequality and threatens democracy. Crown (2017)
  27. Pearl, J.: Direct and Indirect Effects, p. 373–392. Association for Computing Machinery, New York, NY, USA, 1 edn. (2022), <https://doi.org/10.1145/3501714.3501736>
  28. Pennebaker, J.W.: The secret life of pronouns. *New Scientist* **211**(2828), 42–45 (2011). [https://doi.org/https://doi.org/10.1016/S0262-4079\(11\)62167-2](https://doi.org/https://doi.org/10.1016/S0262-4079(11)62167-2), <https://www.sciencedirect.com/science/article/pii/S0262407911621672>
  29. Qu, Q., Liu, Q.H., Gao, J., Huang, S., Feng, W., Yue, Z., Lu, X., Zhou, T., Lv, J.: Gender differences in resume language and gender gaps in salary expectations. *Journal of The Royal Society Interface* **22**(227), 20240784 (2025). <https://doi.org/10.1098/rsif.2024.0784>, <https://royalsocietypublishing.org/doi/abs/10.1098/rsif.2024.0784>
  30. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. *OpenAI blog* **1**(8), 9 (2019)
  31. Ravfogel, S., Twiton, M., Goldberg, Y., Cotterell, R.D.: Linear adversarial concept erasure. In: Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S. (eds.) Proceedings of the 39th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 162, pp. 18400–18421. PMLR (17–23 Jul 2022), <https://proceedings.mlr.press/v162/ravfogel12a.html>
  32. Ravfogel, S., Vargas, F., Goldberg, Y., Cotterell, R.: Adversarial concept erasure in kernel space. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 6034–6055. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Dec 2022). <https://doi.org/10.18653/v1/2022.emnlp-main.405>, <https://aclanthology.org/2022.emnlp-main.405>
  33. Ross, A., Marasović, A., Peters, M.E.: Explaining nlp models via minimal contrastive editing (mice) (2021), <https://arxiv.org/abs/2012.13985>
  34. Rubenstein, H., Goodenough, J.B.: Contextual correlates of synonymy. *Commun. ACM* **8**(10), 627–633 (Oct 1965). <https://doi.org/10.1145/365628.365657>, <https://doi.org/10.1145/365628.365657>
  35. Shah, D.S., Schwartz, H.A., Hovy, D.: Predictive biases in natural language processing models: A conceptual framework and overview. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics (2020). <https://doi.org/10.18653/v1/2020.acl-main.468>, <http://dx.doi.org/10.18653/v1/2020.acl-main.468>
  36. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps (2014), <https://arxiv.org/abs/1312.6034>
  37. Wang, Y., Qiu, X., Yue, Y., Guo, X., Zeng, Z., Feng, Y., Shen, Z.: A natural language counterfactual generation (2024), <https://arxiv.org/abs/2407.03993>

38. Wu, T., Ribeiro, M.T., Heer, J., Weld, D.S.: Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. arXiv preprint arXiv:2101.00288 (2021)