Paper Review

Benjamín Farías Riquelme[1]

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

Abstract. [...]

1 Introduction

Alongside the increase in capacity and influence of algorithms, there is an increase in the concerns and risks of them inadvertedly 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].

In this survey, we present a brief review of bias evaluation and explainability methods, and discuss them through the scope of Prediction Bias Analysis. We will use the term predictive model (in NLP), to refer to any model that takes text as input, and produces a prediction, decision, or classification as output (e.g., toxic text classification or spam filtering). In the same line, an explainability method would be any method designed to provide insight regarding the prediction process, such as which variables are relevant or what information is being processed at a specific layer of the model.

Before deliniating the concept of Prediction Bias Analysis, to ilustrate the risks of unassessed biases in predictive models in NLP, consider the following toy example:

> **Ex1** 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 to a neural network. This network reads an applicant's anonimized resume, and determines if they should be hired or not. The network is trained with the data from the previous processes, so it replicates the same biases. The next time the company is questioned about its 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 who are better qualified for the job.

In this example, the bias is the association of gender with qualification for the job, and it is originated by the use of biased training data. A proper evaluation of bias is neglected, and an undesired bias is perpetuated, or even worsened, under the illusion of "objective" predictions. In order to prevent a model from replicating an undesired bias, it is necessary to identify how the bias if computed by the model.

A substantial corpus of research has been dedicated to the examination of bias in predictibe models in NLP, and in NLP algorithms in general. Most of them can be encapsuled 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 [2,13].

In general, researchs on techniques for bias measurement propose a metric to quantify the presence a specific bias, such as gender or racial bias, in either the model representations or responses. These are referred in the literature as intrinsic and extrinsic metrics, respectively [8,16]. Intrinsic metrics have been found to not correlate with the responses of the model [12], while extrinsic metrics are a subcategory fairness metrics [7], which, in turn, are metrics directly designed to measure the consequences and symptoms of biases (or other problematic elements), rather than the bias itself. We argue that these issues in the analysis of bias may come from an indiscriminate use of the term bias, to refer to two different terms: *Social Bias* and *Prediction Bias* (Figure 1).

1.1 Social Bias and Predicion Bias

To better explain this idea of Social Bias and Predicion Bias, and how they are confounded, we will continue with the hiring model example:

> **Ex2** An organization against gender bias, dertermined to demonstrate that the model used by the company is biased, recolect the resumes of 100 applicants and their results. They find that men were accepted 4 times more than women, even when they had similar background. With this fairness test, the organization is sure that the model favors men over women, however, the company points out a key detail: the resumes are anonimized. The model does not have any information regarding gender, and thus cannot be biased. The organization examinates the applications again, this time focusing on the diference 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[1]. After editing the resumes to have a similar style and passing them to the model, it is found that the rejection rate is now equitative between men and women, probing that it was, in fact, biased.

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

We have previously stated that, in the example, the bias is the association between gender and qualification for the job. To be precise, this is the Social Bias being replicated by the model. Here, even if a metric indicates that the model's answers are biased, it does not explain why it happens, because it is only measuring the Social Bias in its responses. Only 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 that can be called a Prediction Bias.

We define Social Bias and Prediction Biases as follows:

- **Social Bias:** Prejudices and associations, related to social groups, made by humans. This would be gender discrimination in the example.
- **Prediction Bias:** Association between variables encoded in the operations of the model. This would be the correlation between the variables sentence lenght and word diversity and the outcome in the example.

Social Bias can be a cuase and a consequence of Prediction Bias, but a measurement of Social Bias alone is not enough to correct Prediction Bias.

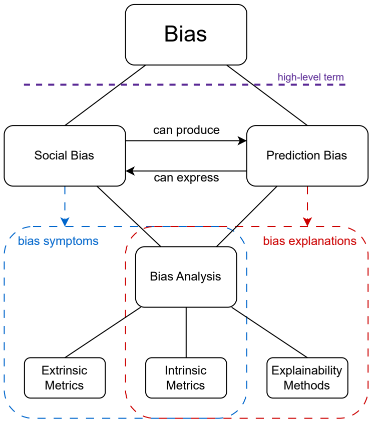

Fig. 1. Use of the term "bias" in bias measurement and explainability literature.

1.2 Prediction Bias Analysis

We indentify an alternative approach for addressing the analysis of bias in predictive models, which could lead to better debiasing methods, in the distinction of Social Bias and Prediction Bias. Rather than measuring the extent to which Social Bias is expressed by a model, we endeavor to identify the Prediction Bias, i.e., the biased associations encoded in the model's prediction mechanism.

In general, we want to reduce a Social Bias expressed by the model, which is caused by a Prediction Bias. However, even if this same Prediction Bias is learned by the model due to the presence of the Social Bias, they are not equivalent. Only proving or measuring the presence of Social Bias in the model's responses does not give enough information regarding the Prediction Bias, and can led to unaccurate assesments or insufficient debiasing. Consider the following alternative version of the example:

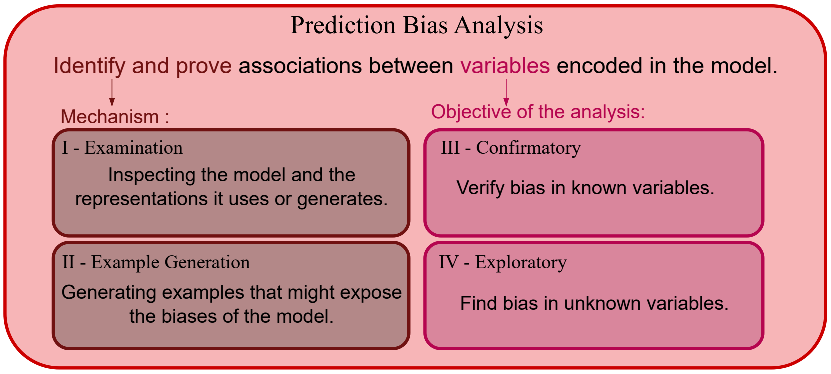
Prediction Bias Analysis
Identify and prove associations between variables encoded in the model.
Mechanism :
Objective of the analysis:
I - Examination
Inspecting the model and the representations it uses or generates.
II - Example Generation
Generating examples that might expose the biases of the model.
III - Confirmatory
Verify bias in known variables.
IV - Exploratory
Find bias in unknown variables.

Fig. 2. ...

Ex3 The company, aware that their model will be accused of perpetuating gender bias, decides to include the gender of the applicants in the input and train the model following a debiasing procedure. The procedure consist on duplicating each resume in the training data, varying only the gender, in order to prevent the model from learning gender bias. Once the training is finished, they test the model and find that gender has no correlation with the output, so the company decides that it is safe to use. Some months later, the company is denounced by the organization, allegating that the model perpetuates gender bias.

In this case, Social Bias and Prediction Bias are confounded. The debiasing method and bias analysis are aimed toward the declared applicant's gender, assuming that the model would learn an explicit association between gender and qualification. However, the model reproduces the bias through a mediator variable. A complete analysis of prediction bias is neglected, failing identify the actual association that cuase the model to expressed social bias.

We denominate the process of indentifying and proving associations in the model, i.e. the exact Prediction Bias, as Prediction Bias Analysis. Even though prediction bias is relevant as already proposed, methods for explicitly addressing Prediction Bias Analysis are scarce. Despite this, we identify an abundance of related methods that, even though some were not specifically designed for this bias evaluation, could be readily adapted for it.

In this survey, we review and discuss the applicability of existing explainability methods in Prediction Bias Analysis. In particular, we consider methods across 5 explainability paradigms: (1) embedding association, (2) concept erasure, (3) ablation, (4) activation maximiztion, and (5) counterfactual examples. In order to constraint the methods into a como ground, where they are comparable, the scope of the survey is limited to methods that can be applied on predictive models with transformer architecture. We chose to center on this architecture of neural network because, at the time of writing, is the most used in state of the art predictive models. Despite of this, some of the methods included were designed, or are applicable, for more general scenarios.

We categorize the methods into 2 groups regarding how they work: (I) Examination methods, that examinates the representations used or generated by the model or its operations; and (II) example generation methods, that generates examples that might show the biases of the model. We also distinguish two different orientations to perform Prediction Bias Analysis, depending on the information available about the variables: (III) Confirmatory Analysis, that seek to verify if there is bias associated to a known variable; and (IV) Exploratory Analysis, that seek to find previously unknown variables related a bias. Figure 2 summarizes the definitions of Prediction Bias Analysis and these four groups.

The remaining of this document is structured as follows: in Section 2.1 we provide further discussion and previously definitions of the concept of Prediction Bias; in Section 3 we review the Examination methods; in Section 4 we review the Example Generation methods; and in Section 5 [...].

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

Table 1. Caption.

2 Background

In this section we review some basic concepts necessary for a better discussion of Prediction Bias Analysis. In Section 2.1 we expand the definition of Prediction Bias with the concept of unintended bias, which is the subcategory of potentially harmful biases.

2.1 Prediction Bias in 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 sunject that will not be thoroughly addressed in this survey. I this work we will limit to the definitions given in Section 1.1, where Social Bias are biases present in the "real world" and Prediction Bias are biased present in the model's computations. Prediction Bias in NLP has been previously described as to the prior that a imforms a predictive model to make its predictions [2,35]. In essence, Prediction Bias correspond to the associations between features of the input and the output of the model. Following this definition, all models have Prediction Bias, and it is not something inherently ploblematic, 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 denomined 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 pattenrs that do not generalizes 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 predictive model in NLP, can be divided into four categories according to the source of the bias [35]:

- **Semantic Bias:** Emerges when the word-embeddings used by the model encode biased relations. *The information represented in the embeddings include a Social Bias (e.g., gender in genderless words).*
- **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. *The training data is affected by a Social Bias, as in in the hiring model example (Ex1).*
- **Selection Bias:** Emerges when the model learns from data that is non-representative of the distribution to where it would be applied. *The training data is selection is affected by a Social Bias (e.g., use mostly texts written by middle- aged white men).*
- **Overamplification:** Emerges when the model itself pick up small difference in the data, and amplify them to be much larger in the predicted outcomes. *The model develops its own unintended Prediction Bias, without the influence of a Social Bias.*

As signalled in the text in cursive, these categories of origin of unintended bias correspond to different forms of how Social Bias can cause Prediction Bias. Identifying the origin of bias can help for the development of unbiased models, however it requieres analysis beyond the boundaries of Prediction Bias Analysis, so we will not cover it in this work.

3 Examination Methods

In this section, we review some methods that can be employed to gain insight on Prediction Bias through the examination of the components of the prediction process, such as the embeddigs or neuron activations. In Section 3.1 we discuss methods of Embedding Association, that compare the vector representations employed by the model; in Section 3.2 we discuss methods for Concept Erasure, that seek to erase specific information from the representations while leaving the rest of it intact; and in Section 3.3 we discuss methods for Model Ablation, that seek to remove parts of the model, such as neurons or conections, without harmimng a given behavior.

3.1 Embedding Association

This category encompas 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. As its name indicates, methods of this family seek to uncover associations between embbedings, with the subjacent objective of identifying information encoded in them.

These are the weakest methods for Prediction Bias Analysis within the scope of this survey. That is because most of them can only confirm that some specific information is present in the embedding, but cannot this information is employed for the model's prediction, or if it employed at all. Despite of this, we decided to include these methods, because they represent a prominent portion of the bias evaluation literature, and because, in theory, the embeddings are ajusted to encode the information that the model needs for its prediction. This means that, if some information is encoded in the embeddigs, then there is a high chance that it is employed for the prediction in some way.

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 stereoty-related bias in word-embeddings. Given 2 sets X, Y of target words (e.g. professions) and 2 sets A, B of attribues words (e.g. gender nouns), WEAT provides a metric (equations 1 and 2) 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.

$$\text{WEAT}(A, B, X, Y) = \frac{\text{mean}_{x \in X}\, s(x, A, B) - \text{mean}_{y \in Y}\, s(y, A, B)}{\text{sd}_{w \in X \cup Y}\, s(w, A, B)} \tag{1}$$

$$s(w, A, B) = \underset{a \in A}{\text{mean}}\, \text{cossim}(w, a) - \underset{b \in B}{\text{mean}}\, \text{cossim}(w, B) \tag{2}$$

WEAT was designed to be applied in contexts where the words of X and Y should be equaly 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 escenarios. 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.

The evaluation performed by WEAT-based methods is closer to a measurement of the extent to which Social Bias is replicated in the embbedings, than to an assessment of Prediction Bias. This is mainly because they do not find any relation between the input and the output of the model. For Prediction Bias Analysis, these methods can be repurposed as a criterion to generate the sets X, Y, instead of evaluating them. For example, solving the double maximization:

$$\max_{X \in V^* \setminus Y} \max_{Y \in V^* \setminus X} \text{WEAT}(A, B, X, Y) \tag{3}$$

where V^* is a target vocabulary. Then these sets can be employed to measure the difference in the model's responses for each of them, which can provide some information regarding the existence of a Prediction Bias related to the association between the sets. This is still a basic analisys, that could be done withput adding WEAT to the equation. Given that WEAT-based methods can only look for predetermined associations, they fall under the Confirmatory Method category.

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 different 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:

$$S_{x,y}(a, b) = \begin{cases} \text{cossim}(x - y, a - b) & \text{if } \|a - b\| \leq \delta \\ 0 & \text{if } \|a - b\| > \delta \end{cases} \tag{4}$$

where δ is a threshold for the distance between a and y.

The uses and limitation of Analogy Generation for Prediction Bias Analysis are basically the same as WEAT, as it designed for establishing associationn between inputs.

3.2 Concept Erasure

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 $C = \{y_i\}_{i=1}^{N}$ that indicates a concept in the vectors (e.g., gender, verb tenses), Concept Erasure methods 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 C.

Concept Erasure offers the input-output relationship assessment that Embedding Association lacks. While Embedding Association can prove the presence of a concept C in the representations, Concept Erasure, by removing C, provides a way to evaluate is the concept is relevant for the model's predictions.

R-LACE Relaxed Linear Adversal Concept Erasure (R-LACE) [31] is a method designed for the task of Concept Erasure in the word-embbedings or inner representations. To erase a concept, R-LACE first finds a subspace $B \subseteq \mathbb{R}^d$ that contains the information of the target concept, within the representations, and then projects the vector representations into the ortogonal complement of B. The subspace B is determined by solving the minmax problem:

$$\min_{\theta} \max_{P} \sum_{i=1}^{N} \mathcal{L}(y_i, g^{-1}(\theta^T P x_i)) \tag{5}$$

where $f_\theta(x) = g^{-1}(\theta^T x)$ is a generalized linear model, with parameters θ and link function g, $\mathcal{L}$ is a loss function, and P is a $d \times d$ ortogonal projection matrix that neutralizes a rank k subspace, with k being an hyper-parameter of the algorithm. R-LACE can be expanded to work with non-linear subspaces, by applying a kernel on f_θ [32].

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, for both Confirmatory Analysis and Exploratory Analysis.

Let X be the inner representations given by some layer of a predictive model M, C a response variable associated to a given concept, and $r(X) = \{r(x_i)\}_{i=1}^{N}$ the resulting vectors after applying R-LACE on X to erase C. R-LACE can be employed for Confirmatory Analysis if C indicates a known features of the input. In this case, if the responses of M over X cannot be replicated for $r(X)$, that would indicate that C has a direct effect on the model's predictions.

To employ R-LACE for Confirmatory Analysis, the response variable C can be defined an indicator of whether a representation x_i is assigned to a given prediction y or not. 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 a Prediction Bias, related to the output y, might be found difference between a and b.

3.3 Model Ablation

The objective of Model Ablation methods is to modify the behavior of a model by removing or nullifying a subset of its components, such as neuron or connections between them. Model Ablation can by employed for Prdiction Bias Analysis following a similar logic to application of Concept Erasure. If a determined behavior of the model, with respect to a set of inputs $\mathcal{D}$, is nullified after ablating a component $G_{\mathcal{D}}$, it can be inferred that $G_{\mathcal{D}}$ computes a Prediction Bias associated to $\mathcal{D}$.

Targeted Edge Ablation Targeted Edge Ablation (TEA) [18] is a technique designed to remove an especific 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 defined 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:

$$\min_{W} \mathcal{L}(G_W, D_{\text{train}}) - \alpha \mathcal{L}(G_W, \mathcal{D}) + \lambda(t) R(W) \tag{6}$$

where G_W is M with a mask W applied over the edges of G, D_{train} is a set of train data, disjoint to $\mathcal{D}$, R is a regularization function, α 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 recieves the following combination of the original value v_A and the ablated value μ_A from node A:

$$w_e v_A + (1 - w_e)\mu_A \tag{7}$$

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

If $\mathcal{D}$ is set to represent a biased behavior (Social Bias), and TEA is capable of effectively removing it from the model, then that would comfirm that M computes the specified bias. Moreover it would indentify the components of M that compute the Prediction Bias. If it is complementes with a method to analize the inner representations, it could help to identify the input's features associated to the bias, by comparing the inner representations before and after the ablation.

4 Example Generation

In this section, we review some methods that can be employed to gain insight on Prediction Bias, through the the generation of examples that produce a determined response on the model. In Section 4.1 we discuss methods for Activation Maximization, that seek to generate inputs that maximize some output logit or neuron activation of the model; and in Section 4.2 we discuss methods for Counterfactual Examples Generation, that generate samples that are similar to a given input while attaining a different outcome.

4.1 Activation Maximization

Activation Maximization methods aim to generate input samples that maximize the activation of some neuron of the model, in order to uncover to which features this neuron is sensible. Given a neuron whose activation is interpretable, such as the prediction layer's logits, Activation Maximization can be employed for Exploratory Analysis, to seek for features of the inputs that are associated with the activation of the target neuron.

Here we review two methods for Activation Maximization: Preferred Input Synthesis [25] and Evolutionary Input Optimization [3]. Both of them generate its examples with a generative model, a neural network $G : Z \to X$ that maps points from a vector space, called latent space, into text. The former following a white-box approach and the later a black-box approach. Despite this difference, their appliability of Prediction Bias Analysis is the same.

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 $M : X \to Y$ and a generative model $G : Z \to X$, denoting by M_h the activation of the target neuron h, PIS generate samples by solving:

$$\arg\max_{z \in Z} M_h(G(z)) - \lambda \|z\| \tag{8}$$

where λ 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 approch, with a zero-order optimization on the latent space of a generative model. Given a predictive model $M : X \to Y$, a generative model $G : Z \to X$, and a target prediction $y \in Y$, MEIO generate samples by solving:

$$\min_{z \in Z} \mathcal{L}(M(G(z)), y) \tag{9}$$

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 idenpdently sampled gaussian noise, which is combined with noised added in the previous itariton (the momentum).

4.2 Counterfactual Examples

In machine learning, a counterfactual explanation is defined as an example that illustrates how a different input would result in a different output [20]. In the context of predictive models for NLP, 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]. These modifications on the input, that are both minimal and enought to change the prediction, can highlight which features of the input are participating in a Prediction Bias. In this section we review three differnt methods for counterfactual example generation: POLYJUICE [38], MiCE [33], and GYC [19].

POLYJUICE The POLYJUICE [38] method employs a fine-tuned GPT-2 [30] model to generate counterfactual examples. The model recieves 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. Setting the perturbation instruction to modify a specific feature, POLYJUICE can be employed to perform a weak Confirmatory Analysis.

MiCE The Minimal Contrastive Editing (MiCE) [33] method generate counterfactual examples via a masked language model, denomined editor model, that is trained to fill the masked spaces in an intput text with tokens, such that the resulting text would obtain a given prediction by a predictive model M. The editor model recieves 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.

In contrast to POLYJUICE, MiCE can be employed mainly for Exploratory Analysis. In particular, given an output variable y and an input text x, such that $M(x) \neq y$, can generate a set token flips that result in the model shifting its prediction to y. These flips might highlight features of the input that M associate with y. However, identifying those features would require an exhaustive qualitative analysis or a very well defined and limited task-dataset framework.

GYC The GYC [19] method generates k counterfactual examples, for a given input text X and a condition C, through controlled text generation. It does not require to train or fine-tune a model. The method consists in modelating the distribution $p(\tilde{y}|X, C)$, for next token prediction, 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:

$$\begin{aligned} o_t, H_t &= \mathrm{LM}(y_{t-1}, H_{t-1}) \\ y_t &\sim \mathrm{Categorical}(o_t) \end{aligned} \tag{10}$$

where H_{t-1} is the history matrix that captures the dependency of y_{t-1} on past tokens and o_t are the output logits, that generate the categorical distribution from which y_t is sampled. To generate a text conditioned on C, H is perturbed two times. The first perturbation is to create $\tilde{H}_t$, which enforces the reconstruction of X, and next is to create $\hat{H}_t$, which enforces the condition C. The perturbations are determined by minimizing 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 assosiated to the condition and the diversity loss maximizes the entropy of the generated logits.

GYC can be employed for both Confirmatory Analysis and Exploratory Analysis, depending on how the condition C is defined. If C enforces the aparition of a certain feature, it can be used to measure the effect of this feature on the output. If C enforces a behavior of the model, it can be used to find features that trigger that behavior. For the Exploratory Analysis, it has the advantage over methods like MiCE that C can enforce more complex conditions that just attaining a given output, such as not inserting already known biases[2]. This reduces the extent to which qualitative analysis is needed to identify the biases.

5 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)

[2] e.g., not inserting insults if we are looking for unintended bias in toxic comment classification.

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/
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
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
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/
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 chapter 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
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://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/ravfogel22a.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)