Reverse Soft Counterfactuals for Bias Exploratory Analysis in Language Models

Benjamin Farias Riquelme[1], Felipe Bravo-Marquez[1,2,3] and Valentin Barriere[1,2]

[1]Department of Computer Science, Universidad de Chile, Santiago, Chile
[2]National Center of Artificial Intelligence (CENIA), Santiago, Chile
[3]Instituto Milenio Fundamento de los Datos (IMFD), Santiago, Chile

Abstract

Most mechanisms for bias evaluation in NLP involve the verification of a given hypothesis concerning the bias. Consequently, biases of unknown nature may be neglected. To prevent this issue, the evaluation of bias can be improved through the incorporation of a preliminary exploratory analysis, which can facilitate the formulation of hypotheses regarding unexpected biases. In this work, we propose ReSCE (Reverse Soft Counterfactual Explorer), a method for bias exploratory analysis in language models. ReSCE generates soft counterfactual examples following a reverse causal inference approach through iterative interventions on the word embeddings. These interventions progressively transform an input text into another that is more favorable for the prediction of a given output y. The generated examples can then be used as a data pool to observe the model's biases regarding y. To evaluate ReSCE, experiments were conducted across two case studies, involving two different models and datasets. The findings revealed that the words extracted through the examples are indeed related to the model's biases, as they increase the probability of predicting one output.

Keywords

Natural Language Processing, Counterfactual Examples, Bias Exploratory Analysis,

1. Introduction

In order to mitigate the effects of unintended biases in a language model, it is necessary to first identify those biases. Consider the following toy example:

A model M, employed to automatically delete toxic posts online forum, has been found to be arbitrarily classifying posts written by people of a certain demographic, A, as toxic. To correct this behavior, M was submitted to a debiasing procedure, aimed to mitigate the effect of mentions of A. However, after replacing the model with its debiased version, even when M was no longer sensitive to mentions of A, complaints of discrimination persisted. After a second evaluation, it was found that publications containing the word "whereupon", which frequently used by people of group A, were more likely to be classified as toxic.

In this example, the extent to which the model was biased could not be accurately assessed due to a lack of knowledge regarding the variables that trigger the bias. A large number of works have been dedicated to the evaluation of bias in neural networks applied to NLP [1, 2, 3, 4]. However, the majority of them require prior knowledge about the nature of the bias (e.g., a well-defined set of protected attributes). In order to achieve a more effective measurement and mitigation of unintended bias, it is necessary to develop methods for exploratory analysis. Such methods should be capable of identifying previously unknown stimuli for biases, such as the word "whereupon" in the given example.

In this work, we present the Reverse Soft Counterfactual Explorer (ReSCE) algorithm, a method for exploratory analysis of bias for differentiable (or semi-differentiable) models that take word-embeddings as input and return a prediction (e.g., next token prediction, classification, masked language modelling). ReSCE progressively transforms an input text X, through gradient updates on the word-embeddings, into a text that has a higher probability of attaining a target prediction y by a given predictive model f.

AMW 2026: 17th Alberto Mendelzon International Workshop on Foundations of Data Management, November 9th–November 13th, 2026, Arequipa, Peru

bfarias@dcc.uchile.cl (B. F. Riquelme); fbravo@dcc.uchile.cl (F. Bravo-Marquez); vbarriere@dcc.uchile.cl (V. Barriere)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Method	Optimization Space	Decoder	Optimization Mechanism	Counterfactual Type	Objective
SCENE	Input tokens	Masked language model	Replace relevant tokens	Soft counterfactual	Counterfactual explanation
Latent-CF	Autoencoder latent space	Autoencoder's decoder	Gradient descent	Reverse counterfactual	Counterfactual explanation
ReSCE	Input embeddings	Interpreter model	Gradient descent	Reverse soft counterfactual	Bias exploratory analysis

Table 1
Key differences between SCENE, Latent-CF and ReSCE.

This transformation addresses the reverse inference question: "How should X change to be closer to prediction y?".

2. Related Work

2.1. Bias in NLP

There is a lack of consensus regarding the definition of bias [1] in NLP. We will use the term "bias" to denote the prior that informs a model's predictions [5, 2]. According to this definition, bias constitutes an inherent component of a model's mechanism. In this context, the term "unintended biases" is employed to denote harmful or inappropriate precedents [5, 2].

The majority of works on the subject of bias in NLP propose methodologies for measuring or mitigating bias (reducing a bias metric). These metrics quantify the presence of a specific bias in either the model's representations (intrinsic) or responses (extrinsic) [3, 4]. Methods designed to identify unanticipated biases are scarce. Some measurement methods can be utilized to automatically find associations between embeddings [6, 7, 8], however they do not provide any insight into the priors of the model.

2.2. Counterfactuals Explanations

Given a model f and an input text X, in NLP, a counterfactual explanation is defined a as any text X' similar to X, such that $f(X) \neq f(X')$ [9, 10, 11]. Algorithms that generate counterfactual examples are called counterfactual explainers [9].

To the best of our knowledge, the only counterfactual explainer developed for the generation of soft counterfactuals in NLP besides ReSCE is SCENE [12]. However it does not generate them in a progressive manner, and neither for the purpose of bias exploratory analysis. A more thorough delineation of the concept of soft counterfactuals is provided Section 3.1.

2.3. Reverse Causal Inference

Causal inference aims to ascertain the existence of a causal relationship between two variables, and to quantify the magnitude of its effect [13, 14, 15]. Standard causal inference follows a forward approach, which entails the estimation of the effects on a response variable Y that are produced by changes in an input variable X. Conversely, reverse causal inference aims to ascertain the alterations in X that yield a specific value of Y [13, 16].

The utilization of reverse counterfactuals, in the context of machine learning, has been previously proposed by Balasubramanian et al. [17] with Latent-CF. Latent-CF is a counterfactual explainer that generates reverse counterfactuals by optimizing values in the latent space of an autoencoder. The authors evaluated the algorithm on images and tabular data. To the best of our knowledge, no other method had been proposed to explicitly generate reverse counterfactuals in NLP. Table 1 presents a summary of the key distinctions between SCENE, Latent-CF and our algorithm.

3. Soft Counterfactuals and ReSCE

In this section we introduces the concept of soft counterfactual examples along with the Reverse Soft Counterfactual Explorer (ReSCE), the algorithm that generates them. ReSCE generates consecutive

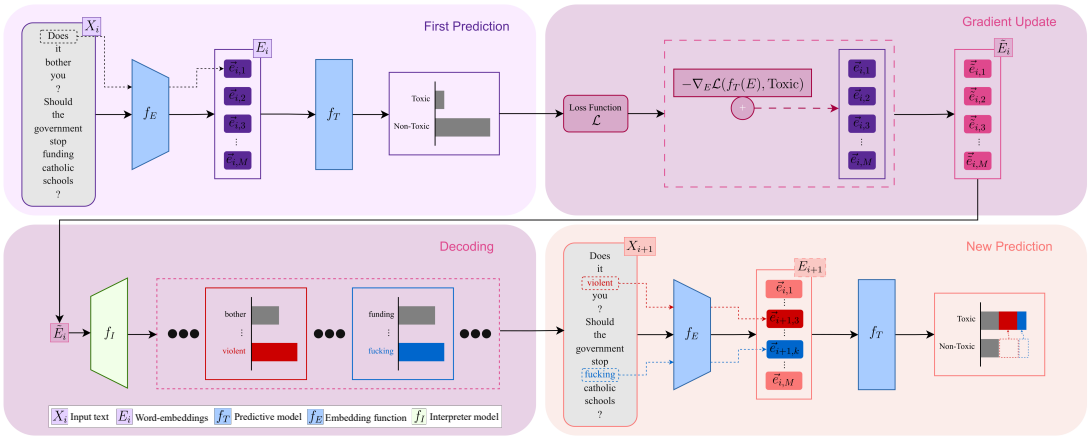

Figure 1: Diagram of ReSCE. The displayed text corresponds to a shortened version of a sample from Civil Comments.

soft counterfactuals for exploratory bias analysis, by progressively transforming an input text into one more favorable for a target prediction y. This process is conducted without imposing conditions such as grammatical coherence or plausibility, in order to prioritise to the stimuli most related to the model's predictions.

3.1. Soft Counterfactual Examples

Soft counterfactuals are counterfactuals where the alternative outcome produced by the modifications lies in the probability of a particular output rather than in the output itself. This change of focus facilitates the quantification of the extent to which a counterfactual modification is relevant to the prediction of a particular output y. Consequently, these examples can be employed to identify of words or patterns in the input that have a higher influence on the prediction of y.

The definition of counterfactual examples can be expanded for soft counterfactuals by replacing the constraint $f(X) \neq f(X')$ for a difference in the outcome's probability, $|P(f(X) = y) - P(f(X') = y)| \geq \varepsilon$, where ε is some reasonable threshold.

3.2. Counterfactual Generation Process

Let $f : V^M \to \mathbb{R}^n$ be an NLP model, where V is the vocabulary and M is the length of the input sentence. The model can be decomposed as $f = f_T \circ f_E$, with f_E being the embedding function, that transforms tokens $x \in V$ to their vector representations $e \in \mathbb{R}^d$, and f_T the remaining layers, which fulfill some task T, such as language modeling or sequence classification.

In order to generate the counterfactuals, the original input tokens, X_0, are transformed by f_E to their respective embedding representations, E_0. Then, at each step i, a new candidate, $\tilde{E}_i$, is computed by subtracting the normalized gradient of the loss, $\mathcal{L}(f_T(E_{i-1}), y)$, to the previous embeddings, with a step size given by $\lambda(M) = \alpha M + \beta$ [1].

The vectors $\tilde{E}_i$, which can be seen as noisy embeddings, are converted back to tokens, X_i, by the interpreter f_I. If any token changed in this step, i.e., $X_i \neq X_{i-1}$, they are transformed to their right embeddings, $E_i = f_E(X_i)$, before passing to the next iteration. Otherwise, the vectors $\tilde{E}_i$ are kept. ReSCE's pseudo-code is provided in Algorithm 1, and Figure 1 displays a diagram of an iteration of ReSCE.

[1]The values $\alpha, \beta \in \mathbb{R}_+$ are hyper-parameters of ReSCE.

Algorithm 1 ReSCE

$f_E \leftarrow$ embedding layer
$f_T \leftarrow$ task layers
$f_I \leftarrow$ interpreter model
$y \leftarrow$ target (predicted token, class label, etc.)
$X_0 \leftarrow [x_{0,1}, x_{0,2}, ..., x_{0,M}] \leftarrow$ input
$E_0 \leftarrow [e_{0,1}, e_{0,2}, ..., e_{0,M}] \leftarrow f_E(X_0)$
for $i \leftarrow 1$ to N **do**
  $g \leftarrow \nabla_E \mathcal{L}\left(f_T(E_{i-1}), y\right)$
  $\tilde{E}_i \leftarrow E_{i-1} - \lambda(M)\frac{g}{\|g\|}$
  $X_i \leftarrow f_I(\tilde{E}_i)$
  if $X_i \neq X_{i-1}$ **then**
    $E_i \leftarrow f_E\left(X_i\right)$
  else
    $E_i \leftarrow \tilde{E}_i$
  end if
end for

3.3. The Interpreter Model

The interpreter model, denoted here by f_I, is a neural network that translates values in $\mathbb{R}^d$ into tokens according to how f_T "understands" them. Given a perturbed embedding $\tilde{E}_i$, the function of f_I is to ascertain the tokens whose embeddings would engender the most analogous response on f as $\tilde{E}_i$ on f_T. To achieve this, f_I is built as a shallow network attached to the latest layers of f_T, before the prediction head, and is trained to reconstruct the tokens read by the model. It is worth noting that the changes decoded by f_I occur at the level of the model's inner representations, rather than within the embedding level. In fact, the analysis conducted revealed that the nearest valid embedding to $\tilde{E}_i$ is E_{i-1}.

The interpreter adds the component of reverse inference to ReSCE. Each step of ReSCE can be understood as an unrealistic intervention over the embeddings, $E \rightarrow \tilde{E}_i$, where $\tilde{E}_i$ is a vector that does not correspond to the embeddings of any tokens. Then, f_I performs the reverse inference task of seeking a valid input X_i, that could produce a response similar to the intervened embeddings.

4. Experiments

The ReSCE algorithm was developed as a tool for exploratory analysis of bias in NLP predictive models. As such, it is challenging to establish a benchmark to evaluate its performance. For this reason, we evaluated ReSCE across two case studies, one of masked language modelling (MLM), with BERT [18] and one of toxic comment classification (TCC), with a fine-tuned GPT-2 model [19]. For MLM, we used the dataset WinoBias [20] which contains around 1,500 sentences depicting an interaction between two persons. We masked the pronouns of each of them, and set ReSCE to increase the probability of BERT predicting male and female pronouns. For TCC, we used the dataset Civil Comments (CC) [21], which has around 2 million comments from social media, and set ReSCE to increase the probability of GPT-2 classifying them as toxic and non-toxic.

In both cases, the algorithm was set to execute ten iterations for each sample, and the hyper-parameters α and β were set to the values of 0.0177 and 0.41, respectively. A large value of $\lambda(M)$ would displace the embeddings farther than intended, resulting in senseless text. Conversely, a too small value would take several iterations to change a token. Taking this into account, we tested a series of 1000 random values for α and β, between 0 and 1, and selected the smaller ones that yield at least one token shift in at least 90% of the samples.

4.1. Soft Counterfactuals Generation

In general, the modified texts, X_{i+1}, do not differ considerably from their previous versions X_i. In the majority of cases, less than the 11% of the tokens were modified in each step. To measure semantic closeness preservation, we defined relative closeness as: $RC(X_i, X_j) = \frac{Sim(X_i,X_j)-\mathbb{E}(Sim_t)}{1-\mathbb{E}(Sim_t)}$, where t is the number of different tokens between X_i and X_j, and $\mathbb{E}(Sim_t)$ is the expected similarity with t random token substitutions. In both case studies, the relative closeness between the original input and its best counterfactual reached values between 0.4 and 0.5, a decrease 2 to 2.5 times below the expected value (see Table 4 in the Appendix).

A notable increase in the probability of the target prediction was attained in over the 95% of the instances, and, in 75% of them, the prediction was shifted to the target. On average, each step of ReSCE increases the target's probability by 0.1 to 0.2 prior to the prediction shift. After the shift, in two-thirds of the instances, the probability undergoes a decrease, which is consistent with the notion of proximity to a local optimum (see Table 5 in the Appendix).

4.2. Bias Exploratory Analysis

In this section, we propose a simple methodology for employing ReSCE's counterfactuals for bias exploratory analysis, and present its results for both case studies.

4.2.1. Common Positives and Negatives

Once the counterfactuals have been generated, the simplest evaluation that can be done for identifying biases, yet not less informative, is to register all the tokens that, when inserted into an input, result in an increase in the target's probability. We denominate this set of tokens *common positives*. The term comes from the fact that these tokens attain relevance when their frequency is taken into account. If a token is found to be frequently added in a wide number of counterfactuals, this might indicate that it corresponds to a concept that the model associates with the target.

Symmetrically to common positives, we also define the *common negatives*, as the set of frequently deleted tokens. While common positives aid to identify words related to the target, common negatives aid to identify words that deviate the model's predictions from it. This can be caused because these words are negatively associated to the target, or because they increase another prediction's probability.

4.2.2. Relevance Score

In order to quantify the importance of common positives and negatives, we attached to each of them a score given by $S_{+/-}(w) = \frac{\sum_{\Delta P \in W_{+/-}} \Delta P}{|C_{+/-}|} \cdot 100$, where w is a token, $C_{+/-}$ is the set of unique common positives/negative, and $W_{+/-}$ are the increases in the target's probability of each counterfactual where w was added/removed. Tables 2 and 3 display the top-10 common positives and negatives for each prediction for WinoBias and Civil Comments, respectively.

In order to assess the correlation between $S_{+/-}$ and the effect that the corresponding token has on the target's probability, we masked one of the two nouns in each sentence in WinoBias. We then filled the mask position with the common positives and negatives, and computed the target's probability increase with respect to the masked sentences (see Figure 2 in the Appendix).

The Pearson correlation coefficient, between the mean probability increase and S_+, is 0.499 for female pronouns and 0.194 for male pronouns. Concurrently, the coefficients for S_- are -0.093 for female pronouns and -0.216 for male pronouns. The scores $S_{+/-}$ do not exhibit a high correlation with the target's probability increase. Nevertheless, the information provided is sufficient for its intended purpose, which is to create a set of tokens that facilitates the formulation of hypotheses concerning the biases of the model.

Among the common positives for WinoBias, there are words, such as "nurse" or "doctor", that correspond to known gender stereotypes [22]. It can be seen that stereotype-related words appear

more often for female pronouns common positives and male pronouns common negatives. This, in conjunction to the fact that the probability increase for female pronouns is more corrected with common positives and for male pronouns is more corrected with common negatives, suggest that the prediction of female pronouns may be conditioned by the presence of female-related words (including stereotypes), while the male counterpart tends to be the default prediction. A similar scenario is observed for TCC, where the values of S_+ for toxic prediction are found to be considerably higher than those for non-toxic prediction.

5. Conclusion

After testing ReSCE in two case studies, we found that our mechanism of inserting gradient updates on the embeddings and decoding the latter representations into tokens achieves the desired optimization over the input. In other words, the introduction of minor alterations, both in terms of sentence length and semantic level, on an input X, that result in an increase of the probability assigned to an output y. The generated samples are useful for bias exploratory analysis. In the case studies, it can be observed that BERT favors the prediction of male pronouns, unless a female-biased word is present in the text, and that toxic classification depends mostly on the apparition of certain words, such as "stupid" or "idiot". These observations are consistent with the already established knowledge about gender and toxicity bias, which demonstrates that ReSCE is capable of exposing biases of the model. Consequently, in the initial phases of the analysis, when the nature of the biases is unknown, ReSCE could be a useful tool for bias exploration, as it does not require instructions regarding how to modify X.

Our results show that the algorithm is capable of finding biases without prior knowledge, but we cannot say much about how much "unexpected" these findings could be for human evaluators. For future work, an evaluation of the difference between the biases expected by humans and found biases, similar to the "surprise score" defined by Wu et al. [23], might help to answer this question.

Declaration on Generative AI

During the preparation of this work, the author(s) used DeepL Write in order to: Grammar and spelling check.

References

[1] S. L. Blodgett, S. Barocas, H. Daumé III, H. Wallach, Language (technology) is power: A critical survey of "bias" in NLP, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 5454–5476. URL: https://aclanthology.org/2020.acl-main.485/. doi:10.18653/v1/2020.acl-main.485.

[2] D. S. Shah, H. A. Schwartz, D. Hovy, 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. URL: http://dx.doi.org/10.18653/v1/2020.acl-main.468. doi:10.18653/v1/2020.acl-main.468.

[3] P. Czarnowska, Y. Vyas, K. Shah, Quantifying social biases in nlp: A generalization and empirical comparison of extrinsic fairness metrics, Transactions of the Association for Computational Linguistics 9 (2021) 1249–1267.

[4] V. Gupta, P. Narayanan Venkit, S. Wilson, R. Passonneau, Sociodemographic bias in language models: A survey and forward path, in: A. Faleńska, C. Basta, M. Costa-jussà, S. Goldfarb-Tarrant, D. Nozza (Eds.), Proceedings of the 5th Workshop on Gender Bias in Natural Language Processing (GeBNLP), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 295–322. URL: https://aclanthology.org/2024.gebnlp-1.19/. doi:10.18653/v1/2024.gebnlp-1.19.

[5] R. Bansal, A survey on bias and fairness in natural language processing, 2022. URL: https://arxiv.org/abs/2204.09591. arXiv:2204.09591.

[6] A. Caliskan, J. J. Bryson, A. Narayanan, Semantics derived automatically from language corpora contain human-like biases, Science 356 (2017) 183–186. URL: https://www.science.org/doi/abs/10.1126/science.aal4230. doi:10.1126/science.aal4230. arXiv:https://www.science.org/doi/pdf/10.1126/science.aal4230.

[7] W. Guo, A. Caliskan, 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, AIES '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 122–133. URL: https://doi.org/10.1145/3461702.3462536. doi:10.1145/3461702.3462536.

[8] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, A. T. Kalai, Man is to computer programmer as woman is to homemaker? debiasing word embeddings, in: D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 29, Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper_files/paper/2016/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf.

[9] R. Guidotti, Counterfactual explanations and how to find them: literature review and benchmarking, Data Mining and Knowledge Discovery (2022) 1–55.

[10] A. Madsen, S. Reddy, S. Chandar, Post-hoc interpretability for neural nlp: A survey, ACM Comput. Surv. 55 (2022). URL: https://doi.org/10.1145/3546577. doi:10.1145/3546577.

[11] Y. Wang, X. Qiu, Y. Yue, X. Guo, Z. Zeng, Y. Feng, Z. Shen, A natural language counterfactual generation, 2024. URL: https://arxiv.org/abs/2407.03993. arXiv:2407.03993.

[12] H. Zheng, U. Pamuksuz, Scene: Evaluating explainable ai techniques using soft counterfactuals, 2024. URL: https://arxiv.org/abs/2408.04575. arXiv:2408.04575.

[13] J. S. Mill, A system of logic, 1843.

[14] P. W. Holland, Causal inference, path analysis and recursive structural equations models, ETS Research Report Series 1988 (1988) i–50.

[15] J. Pearl, Causality, Cambridge university press, 2009.

[16] A. Gelman, G. Imbens, Why ask why? Forward causal inference and reverse causal questions, Technical Report, National Bureau of Economic Research, 2013.

[17] R. Balasubramanian, S. Sharpe, B. Barr, J. Wittenbach, C. B. Bruss, Latent-cf: A simple baseline for reverse counterfactual explanations, 2021. URL: https://arxiv.org/abs/2012.09301. arXiv:2012.09301.

[18] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[19] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI blog 1 (2019) 9.

[20] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, K.-W. Chang, Gender bias in coreference resolution: Evaluation and debiasing methods, 2018. URL: https://arxiv.org/abs/1804.06876. arXiv:1804.06876.

[21] D. Borkan, L. Dixon, J. Sorensen, N. Thain, L. Vasserman, Nuanced metrics for measuring unintended bias with real data for text classification, CoRR abs/1903.04561 (2019). URL: http://arxiv.org/abs/1903.04561. arXiv:1903.04561.

[22] M. Kaneko, D. Bollegala, Debiasing pre-trained contextualised embeddings, 2021. URL: https://arxiv.org/abs/2101.09523. arXiv:2101.09523.

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

Female		Male		Female		Male	
Token	S_+	Token	S_+	Token	S_-	Token	S_-
Nurse	16.701	Doctor	2.915	Driver	8.139	Nurse	10.979
Receptionist	4.582	Director	2.041	Designer	7.019	Housekeeper	9.233
Police	3.219	Architect	1.625	Manager	7.013	Receptionist	7.569
Cook	2.318	Farmer	0.977	Doctor	6.554	Cook	4.034
Singer	1.957	Young	0.976	Editor	4.906	Baker	3.736
Designer	1.862	Resident	0.973	Writer	4.865	Teacher	3.600
Housekeeper	1.827	Sheriff	0.921	Baker	4.754	Cleaner	3.599
Victim	1.529	Driver	0.822	Tailor	3.978	Secretary	3.405
Secretary	1.517	Stranger	0.738	Clerck	3.974	Designer	3.351
Beauty	1.111	Player	0.698	Farmer	3.849	Counselor	3.167

Table 2
Top genderless nouns common positives (S_+) and negatives (S_-) for pronoun prediction in WinoBias.

Toxic		Non-Toxic		Toxic		Non-Toxic	
Token	S_+	Token	S_+	Token	S_-	Token	S_-
Stupid	81.452	People	1.040	Disgusting	0.458	Stupid	0.659
Idiot	40.897	Y	0.397	Old	0.395	Dumb	0.460
Idiots	24.650	Http	0.361	Racist	0.373	Crap	0.347
Crap	21.912	Trump	0.355	Racists	0.362	Idiots	0.319
People	11.893	Able	0.338	Want	0.303	Hells	0.298
Trump	9.906	Americans	0.267	Black	0.283	Idiot	0.285
Countries	9.724	Time	0.257	Gay	0.266	Gay	0.232
Ass	7.825	Man	0.256	Ignorant	0.265	Fool	0.193
Time	5.609	Old	0.251	Ate	0.250	Suck	0.180
America	5.000	Ant	0.222	All	0.238	Racist	0.156

Table 3
Top nouns, adjectives, verbs and proper names common positives (S_+) and negatives (S_-) for toxic classification in Civil Comments.

A. Common Positives and Negatives Tables

B. Soft Counterfactual Generation Results

Tables 4 and 5 report detailed quantitative results for the two case studies described in Section **??**. Table 4 summarizes text modification metrics, measuring how much the counterfactuals diverge from their source texts. Table 5 reports the probability increase metrics, capturing the effectiveness of ReSCE at shifting the target prediction.

	Mean F_s	Mean F_c	Mean RC_s	Mean RC_h
WinoBias (BERT)	10.5%	14.7%	0.257	0.465
Civil Comments (GPT-2)	4.9%	17.88%	0.739	0.489

Table 4
Text modification metrics. F_s is the fraction of text modified in each step; F_c is the fraction of text modified to change the prediction; RC_s is the relative closeness between two consecutive counterfactuals; and RC_h is the relative closeness between the original text and the example with highest target's probability.

	Mean P_s^*	Mean P_{up}	Mean P_{max}	N_c
WinoBias (BERT)	0.129	0.397	0.738	72.02%
Civil Comments (GPT-2)	0.201	0.293	0.792	78.5%

Table 5
Target's probability increase metrics. P_s^* is the step probability increase before the prediction shift; P_{up} is the probability difference between the initial text and the best sample; P_{max} is the probability of the best sample; and N_c is the percentage of cases where a prediction shift was achieved.

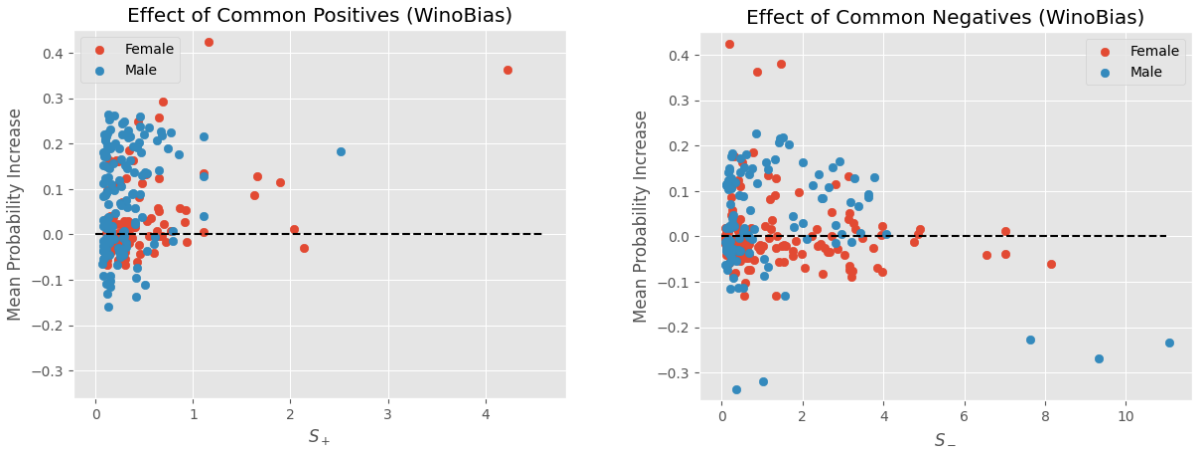

Figure 2: Mean increase in the target prediction's probability produced by inserting common positives (left) and negatives (right) in the sentences of WinoBias.

C. Relevance Score Correlation

Figure 2 shows the mean increase in the target prediction's probability obtained by inserting common positives (left) and common negatives (right) at a masked noun position in WinoBias sentences. Each point corresponds to one token, and its horizontal coordinate is its relevance score $S_{+/-}$. The moderate Pearson correlations (0.499 for female S_+, -0.216 for male S_-) confirm that the score serves as a useful, if imperfect, proxy for a token's causal effect on the prediction.