NoMatterXAI: Generating "No Matter What" Alterfactual Examples for Explaining Black-Box Text Classification Models

Tuc Nguyen[1], **James Michels**[2], **Hua Shen**[3], **Thai Le**[1]

[1] Department of Computer Science, Indiana University
[2] Department of Computer Science, University of Mississippi
[3] Information School, University of Washington
tucnguye@iu.edu, jrmichel@go.olemiss.edu, huashen@uw.edu, tle@iu.edu

Abstract

In Explainable AI (XAI), counterfactual explanations (CEs) are a well-studied method to communicate feature relevance through contrastive reasoning of *"what if"* to explain AI models' predictions. However, they only focus on important (i.e., relevant) features and largely disregard less important (i.e., irrelevant) ones. Such *irrelevant features* can be crucial in many applications, especially when users need to ensure that an AI model's decisions are not affected or biased against specific attributes such as gender, race, religion, or political affiliation. To address this gap, the concept of *alterfactual explanations* (AEs) has been proposed. AEs explore an alternative reality of *"no matter what"*, where irrelevant features are substituted with alternative features (e.g., "republicans"→"democrats") within the same attribute (e.g., "politics") while maintaining a similar prediction output. This serves to validate whether the specified attributes influence AI model predictions. Despite the promise of AEs, there is a lack of computational approaches to systematically generate them, particularly in the text domain, where creating AEs for AI text classifiers presents unique challenges. This paper addresses this challenge by formulating AE generation as an optimization problem and introducing NoMatterXAI, a novel algorithm that generates AEs for text classification tasks. Our approach achieves high fidelity of up to 95% while preserving context similarity of over 90% across multiple models and datasets. A human study further validates the effectiveness of AEs in explaining AI text classifiers to end users. The code is available at https://github.com/nguyentuc/NomatterXAI.

1 Introduction

As AI advances, complex machine learning (ML) text classifiers have been developed to yield predictive performance competitively to that of humans for myriad tasks (Pouyanfar et al. 2018). However, many of such models are so-called "black-box" models that are notorious for their lack of transparency. This may limit both the comprehension and societal acceptance of ML in critical fields, such as healthcare (Tjoa and Guan 2021), finance (Benhamou et al. 2021), and content moderation (Kemp and Ekins 2021). The field of Explainable Artificial Intelligence (XAI) (Adadi and Berrada 2018) aims to remedy this by explaining the factors at play in a model's predictions.

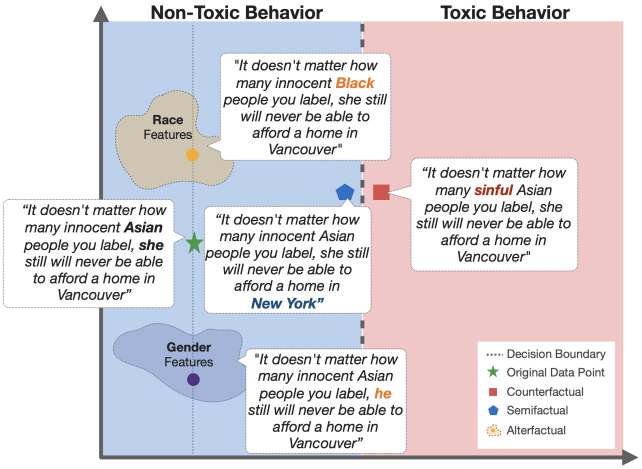

Figure 1: A comparison of various AI explanation algorithms, including Counterfactual, Semifactual, and our proposed Alterfactals explanations. Alterfactual explanations aim to validate whether AI model predictions are influenced by specific attributes such as race or gender.

A common paradigm found in XAI is the counterfactual explanation (CE) (Miller 2019) where an alternative reality is presented where minor alterations to input directly change an AI model's output applied to image, tabular, and text data classification problems (Verma, Dickerson, and Hines 2020; Garg et al. 2019; Yang et al. 2020). CE follows the thought process of counterfactual thinking by asking "***What if...?***", which is a common occurrence in the human psyche, through emotions such as regret, suspense, and relief (Roese and Morrison 2009). CE is often delivered via natural language in the form of "What if" messages (Le, Wang, and Lee 2020; Hendricks et al. 2018). For example, a classifier that labels email messages as spam or ham could provide the text "*Had* the word 'credit' and 'money' is used twice in the message, it would have been classified as spam *rather than* ham." (Le, Wang, and Lee 2020).

While CE is highly effective at providing intuitive reasoning to the users by emphasizing important features, it often neglects the role of less important ones in a text input, *occluding information on what is indeed irrelevant to a model's decision*. However, in many cases, irrelevant features are as important as relevant ones in explaining black-box predic-

tions. For example, irrelevant features can help (1) contribute to the comprehensive understanding of a black-box model (Mertes et al. 2022) and (2) determine whether a model is biased against specific semantic features such as gender or race, which we cannot fully understand with only CE.

A recent study posed a solution in the form of ***alterfactual explanation*** (AE) (Mertes et al. 2022). AEs embody the thought process of ***"No matter what..."*** and present an alternative reality where a set of irrelevant features are *significantly changed*, and yet the model's output remains the same. While (Mertes et al. 2022) demonstrate that users view AEs equally favorably as counterfactual explanations, this was done with a hypothetical model for tabular data presented in the user study. The algorithmic generation of AEs for actual trained models is still needed. This can be achieved for tabular data by changing individual features significantly up to their domain ranges–e.g., alternating "age" of a patient from 0 to 100. However, in the NLP domain, textual features cannot be as directly altered due to their discrete nature, not to mention how to change a textual feature significantly but still maintain the reasonable semantic context of the original input–e.g., changing "Republicans"→"Democrats" as shown in Fig. 1 is non-trivial. Thus, not only has the generation of AEs for text classifiers not been explored, but such a task also has its unique challenges.

As a first step to exploring AEs for text classification tasks, this work investigates how to systematically generate alterfactual examples for text classifiers. We propose a framework, called NOMATTERXAI, that can *significantly change different irrelevant features* of an input text to generate alterfactual for a target ML classification model. Our contributions are summarized as follows.

1. We elucidate a formal definition of AEs for text data. This definition is in an ideal theoretical form, and we explicate how it is translated to our solution.
2. We introduce a novel algorithm NOMATTERXAI, which generates alterfactual variants of input texts, such that one or more irrelevant words are changed by opposite words selected via two strategies, ConceptNet and ChatGPT while maintaining almost no noticeable changes in prediction probability and original context similarity.
3. We conduct both automatic and human evaluations on four real-world text datasets, three text classifiers, achieving up to 95% in effectiveness of generating AEs, showing that such AEs can support humans to accurately compare biases among different classification models.

2 Background and Motivation

This section provides a summary of a variety of factual explanation examples applied to the NLP domain, including *semifactuals*, *counterfactuals*, and *adversarials*. This will help distinguish the alterfactual from the rest (Table. 1).

Counterfactuals are shown to be intuitive to humans by explaining *"Why X, rather than Y"* for a model's decision such as *"This email would be classified as ham rather than spam if there were 50% less exclamation points"* (Le, Wang, and Lee 2020). Counterfactual explanations (CEs) are traditionally used in classification tasks (Verma, Dickerson, and Hines 2020) and recently information retrieval tasks (Kaffes, Sacharidis, and Giannopoulos 2021; Agarwal et al. 2019; Tan et al. 2021). They tend to be minimal such that the input is perturbed *as little as possible* to yield a contrasting output (Kenny and Keane 2021).

Type	Example
Factual	**Since** your income is $100K, you get the loan
Semifactuals	**Even if** your income is $80K, you get the loan
Counterfactuals	**If** your income was $1K lower, you **would had** not got the loan
Alterfactuals	**No matter what** your *race* is, you would get the loan with your current income

Table 1: Examples of different types of explanations in a hypothetical scenario where an algorithm determines whether or not a person is approved for a loan based on their income.

Semifactuals explain *"Even if X, still P."*, or that an identical outcome occurs despite some *noticeable* change in the input, explaining such as *"This email is still spam even if it had 3 exclamation marks instead of 6"*. The exact definition varies, either as an input that is modified to be closer to the decision boundary (Kenny and Keane 2021) or others consider any input of the same class to be semifactual (Kenny and Huang 2023).

Adversarials result from slight alterations to an input designed to fool an ML model's prediction. While closely related to counterfactual explanations (CEs) (Le, Wang, and Lee 2020), adversarial examples differ in their intent—i.e., to confuse a model rather than provide interpretability. They are similar to CEs in that they involve minimal changes intended to yield a different classification. However, adversarial attacks are typically crafted to be imperceptible to humans, whereas counterfactuals are meant to be detectable and interpretable by humans.

Alterfactuals as defined by Mertes et al. (2022), is a variant of semifactuals:

DEFINITION 1. Alterfactual Example in ML. Let denote d be a distance metric on input space X, $d : X \times X \rightarrow R$. An alterfactual example of an example x with a model M is an altered version $x^* \in X$, that *maximizes the distance* $d(x, x^*)$ with the distance to the decision boundary B and the prediction of the model do not change–i.e., $d(x, B) \approx d(x^*, B)$ and $f(x)=f(x^*)$.

Motivation. While CEs present scenarios where negligible changes can alter an outcome, they focus less on identifying which features are irrelevant. Because feature changes in CEs are minimal, these explanations may fail to capture all the factors influencing a model's decision-making, including both relevant and irrelevant signals. Adversarial explanations (AEs), on the other hand, can highlight irrelevant features by exaggerating their influence (Mertes et al. 2022). This perspective offers a novel and intriguing approach to model explanation. However, Mertes' study primarily measures the effectiveness of AEs in explaining model behaviors to users.

In our work, we aim to examine how AEs can be automatically generated in practice and propose the first method of its kind for the text domain. We refer to the Appendix for a detailed comparison of different types of factual statements.

3 Problem Formulation

Given a sentence x and text classifier M, our goal is to generate new AE x^*, to provide interpretable information on irrelevant features of x of the prediction $f(x)$. According to the *Definition. 1*, we hope to generate AE x^* is changing x as much as possible, or:

$$\max_{x^*} \; d(x, x^*) \tag{1}$$

Moreover, for x^* an alterfactual example, it needs to maintain a similar distance to the decision boundary to the original predicted class and at the same time preserve the original prediction, or:

$$\text{argmax}(f(x^*))=\text{argmax}(f(x)) \wedge |f(x)-f(x^*)| \leq \delta, \tag{2}$$

where δ is a small threshold constraining how much the original prediction probability can shift. However, without any additional constraint, x^* might *not* necessary preserve the same context of x and can even result in meaningless sentences (e.g., "today is monday"→"today is school"). Thus, we want to perturb the original input x (grey circle) to generate optimal x^* that is *also furthest away* from x and x^* to be still within the context space of x, denoted as $\mathcal{S}_x$, or:

$$x^* \in \mathcal{S}_x, \tag{3}$$

However, it remains non-trivial to systematically manipulate an entire sentence x in the *discrete* text space. While manipulating x via its embedding in the continuous vector space is possible, such approaches may produce x^* that is drastically different from x, introducing numerous random changes that are no longer interpretable to users. To address this challenge, we can perturb x through word-level replacements, as commonly done in existing counterfactual explanation (CE) works. By replacing individual words with semantically distant alternatives–e.g., *"pretty" → "ugly"*, we aim to move the entire sentence x as far as possible while maintaining interpretability. We then opt for perturbing only *irrelevant* features x^*_{ir} of x^*. Eq. 1 becomes:

$$\max_{x^*_{\text{ir}}} \; d(x, x^*) \tag{4}$$

Perturbing only the irrelevant features x_{ir} of x provides a more specific and intuitive *"no matter what"* explanation. For instance, an explanation like: *"no matter how we change 'pretty' (e.g., to 'ugly') in the sentence, the prediction remains the same"*. This approach not only ensures interpretability but also increases the likelihood that x^* remains parallel to the decision boundary. In contrast, perturbing relevant or important features is more likely to significantly alter the prediction probability, thereby reducing the utility of the explanation.

Still, we cannot replace x_{ir} with just any perturbation x^*_{ir}. For example, good perturbations include antonyms–e.g., "he"→"she" as in "*no matter what* the gender of the person, the classifier still predicts hate-speech", or members of a distinct group–e.g., "red", "blue", "green" (colors), "democrats", "republicans" (political leaning) as in "*no matter what* the political leaning of the user, the classifier still predicts non-hate-speech'. To enforce this constraint, we require that the replacement token needs to share the same *semantic field* (Jurafsky 2000) with the original one, or:

$$\text{s}(x^*_{\text{ir}}) = \text{s}(x_{\text{ir}}) \; \forall \; x^*_{\text{ir}} \in x^*, \tag{5}$$

where x_{ir} and x^*_{ir} denote arbitrary a pair of original and replacement word and $s(\cdot)$ queries the semantic field of a word. This constraint makes perturbations such as "Monday"→"cool" in "today is Monday and the weather is nice" unfeasible because "cool" and "Monday" does not share the same semantic field, although "cool" is semantically far away from "Monday" and still somewhat preserves the original context. This results in the objective function below.

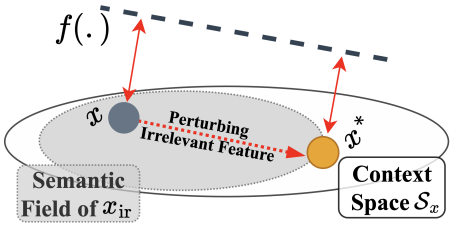

Figure 2: AE generation of x^* (orange circle) from x (grey circle) by perturbing irrelevant features x_{ir} of x within their semantic fields while still maintaining original context of x.

OBJECTIVE FUNCTION: For a given document x with irrelevant features x_{ir}, text classifier M, and threshold hyperparameter δ, our goal is to generate an alterfactual example x^* of x by solving the objective function:

$$\begin{aligned} \max_{\{x^*_{\text{ir}} \in x^*\}} \quad & d(x, x^*) \;\; \text{s.t.} \\ & \arg\max(f(x^*)) = \arg\max(f(x)), \\ & d[f(x) - f(x^*)] \leq \delta, \\ & x^* \in \mathcal{S}_x \\ & \text{s}(x^*_{\text{ir}}) = \text{s}(x_{\text{ir}}) \; \forall \; x^*_{\text{ir}} \in x^* \end{aligned} \tag{6}$$

4 Proposed Method: NoMatterXAI

To solve the objective function, we propose a novel greedy algorithm called NoMatterXAI. Overall, NoMatterXAI involves two steps. Given an input text x, it selects a maximum of m words to perturb in order according to their importance to the prediction $f(x)$. Each word is greedily perturbed with its counterparts while ensuring all the constraints are satisfied. A detailed algorithm is described in the Alg.1.

Step 1: Irrelevant Feature Selection. Each feature of x is ranked from lowest to highest predictive importance based on the probability drop in the original predicted class *when they are individually removed* from x (lines 2-5 in Alg. 1). We prioritize perturbing features of lower importance–a.k.a., irrelevant features, first, since their perturbations are less likely to alter the model's prediction probability to the predicted class. Then, we iteratively transform one word at a time until we have checked a maximum of m words (lines 6-16 in Alg. 1). Hyper-parameter m is set to ensure that (1) there are not too many perturbations in x that could make the resulting AEs difficult to interpret and (2) reduce unnecessary runtime.

Algorithm 1: **AE Generation by** NoMatterXAI

Require: Input sentence $x=\{w_1, w_2, ..., w_n\}$, target model $f(\cdot)$, sentence similarity threshold ϵ, current perturbations p_c, current confidence score c, sentence similarity function $\mathrm{sim}(\cdot)$.
Output: AE x^*, confidence score post perturbation c^*.

Initialize $x^* \leftarrow x$, $i \leftarrow 0$, $p_c \leftarrow 0$, $\delta \leftarrow 0.05$, $c \leftarrow f(x)$
for each word $w_i \in x$ **do**
  Compute the importance score I_{w_i}.
end for
Create a set W of all words $w_i \in x$ sorted by the ascending order of their importance score I_{w_i}
while $i \leq \mathrm{length}(W)$ **do**
  Find antonyms a_i of $W[i]$ by ChatGPT or ConceptNet.
  $x' \leftarrow$ Replace $W[i]$ with a_i in x^*
  $double\_negative\_check = \mathrm{Double\_Negative}(x')$
  if $double\_negative\_check$ == False **then**
    $c' = f(x')$
    cond1 = $|c'$ - $c| \leq \delta$ AND $\mathrm{argmax}(c) == \mathrm{argmax}(c')$
    cond2 = $\mathrm{sim}(x, x') \geq \epsilon$
    if cond1 AND cond2 **then**
      $c^* \leftarrow c'; x^* \leftarrow x'$
    end if
    $i \leftarrow i + 1$
  end if
end while
return (x^*, c^*)

Step 2: Feature Perturbation with Opposite Word. We want to perturb the selected irrelevant features "farthest away" to their originals to move x^* to right at the boundary of $\mathcal{S}_x$ as depicted in Fig. 2. Moreover, such perturbations also need to share the same semantic field of the original token (Eq. 5). We call these opposite words and adopt the definition of *oppositeness* in terms of incompatibility in the linguistic literature–i.e., that is, for example, "if a thing can be described by one of the members of an antonym pair, it can't be described by the other" (Keith 2022). Such opposite words also often share the same semantic field of the original one (Li 2017; Jurafsky 2000).

However, coming up with such perturbations is non-trivial as there is no clear quantitative measure for oppositeness for a word, and most of the relevant literature often desires semantically similar rather than opposite replacements such as in adversarial NLP. Even if we add noise to the original token's embedding to find replacements, it would require very different *bound of noise* for different words to be still in the same semantic field–i.e., the grey region in Fig. 2 is dependent on x_{ir}. For example, the L_2 distance between *Glove* word embeddings (Pennington, Socher, and Manning 2014) between "pretty" and "ugly', "Monday" and "Tuesday", "republicans" and "democrats" are very different: 3.9, 0.4 and 1.3, respectively. Thus, adding a fixed amount much noise might end up in perturbations that are inappropriate.

Therefore, we adopt two different strategies that both leverage external knowledge to find opposite words for replacements: finding perturbations via the ConceptNet database and large language models (LLMs).

Opposite Words Selection via ConceptNet. The selected database for identifying antonymous words is the user-annotated knowledge base ConceptNet (Speer, Chin, and Havasi 2017). ConceptNet's word relations are notably annotated with numerical weightings through various sources. For a transformation of an input word, the following hierarchy of choices is used to identify opposite words (Table. 2).

Method	Example
Original	The children listened to **jazz** all day.
Antonym	The **adults** listened to jazz all day.
DistinctFrom	The children listened to jazz all **month**.
Hyponym	The children listened to **rock** all day.

Table 2: Examples of retrieved opposites from ConceptNet.

- **Antonyms:** ConceptNet's API is called to check for words registered as the input word's antonym via the /r/Antonym relation, such that the weight of the relation is over ω_t.
- **Distinct Items:** ConceptNet's API is called to check for words registered as members of a common set via the /r/DistinctFrom relation, such that something that is A is not B (e.g. red and blue), and that the weight of relation is over ω_t. This ensures that choices of transformed words remain within common groups and can be adequately selected.
- **Hypernym's Hyponym:** We check for an umbrella term, referred to as a hypernym via ConceptNet's */r/IsA relation*, under which the input word belongs. For example, "rose", "lilac" and "iris" are all hyponyms of "flowers". If one is found, a query is made to identify members of the identified category that are not the input word, such that a member of the same category is to be selected. This is intended to identify words that are members of some overarching group, as similarly done in Distinct Items.

Opposite Words Selection via LLM. ChatGPT (OpenAI 2023) estimates the likelihood of subsequent tokens in a text-based on preceding words. We employ the inferred contextual understanding that ChatGPT can offer to identify antonyms. ChatGPT 3.5-Turbo is called for each input sentence and asked to provide one context-relevant antonym per word in the sentence such that the original sentence is still grammatically correct with the antonym replacement. Please refer to the Appendix for full details of the prompt.

Avoiding Double Negatives. When words are changed for antonyms, some words have negative counterparts, such as "is" to "isn't". Multiple of these may cause double-negatives to arise in sentences, which may cause the user-interpreted meaning of the text to not significantly change. To address this, we create a constraint to detect and reject potential double-negative sentences, unless the original text also featured a double-negative. This reduces potential confusing alterfactual texts to be returned to users. Of these replacements, we only keep those that do not create a double negative, do not exchange a word for one that is a different part of speech (ex. noun→verb), and that do not alter the model output confidence score δ beyond 5%. The detailed algorithm is described in Alg. 2 (Appendix).

5 Experiment Settings

This section shows a comprehensive evaluation of NoMatterXAI with different settings and baselines.

	Method	DistilBERT					BERT					RoBERTa				
		FID↑	AWP↑	APPL↓	SIM↑	CON↓	FID↑	AWP↑	APPL↓	SIM↑	CON↓	FID↑	AWP↑	APPL↓	SIM↑	CON↓
GB	Feng et al	**95.56**	**6.65**	156.57	0.81	1.61	**96.58**	**6.44**	154.51	0.83	1.67	**96.98**	**6.45**	153.99	0.84	1.68
	CNet-Single	77.78	1.00	86.41	0.86	1.43	79.44	1.00	86.15	0.86	1.31	78.53	1.00	86.86	0.86	1.33
	GPT-Single	70.83	1.00	**83.90**	0.86	1.36	69.49	1.00	**83.58**	0.86	1.21	67.93	1.00	**83.39**	0.86	1.26
	CNet-Multi	77.78	1.55	98.19	0.88	**1.19**	79.44	1.58	98.05	0.87	1.07	78.55	1.57	94.29	0.87	**1.10**
	GPT-Multi	70.83	1.56	98.29	**0.89**	1.24	69.49	1.59	99.30	**0.89**	**1.04**	67.93	1.57	98.01	**0.89**	1.15
HS	Feng et al	**99.17**	**7.87**	89.63	0.82	0.55	**98.77**	**7.77**	87.90	0.84	0.36	**99.18**	**7.84**	90.93	0.85	0.36
	CNet-Single	92.26	1.00	76.18	0.87	0.47	92.21	1.00	76.49	0.87	0.36	92.29	1.00	**77.25**	0.87	0.25
	GPT-Single	80.14	1.00	**75.71**	**0.88**	0.44	79.32	1.00	**75.94**	0.88	0.32	84.87	1.00	90.84	**0.91**	0.24
	CNet-Multi	92.26	2.34	88.48	0.84	**0.33**	92.21	2.40	90.09	0.84	**0.27**	92.29	2.46	89.62	0.85	**0.18**
	GPT-Multi	80.28	2.24	87.06	0.87	0.37	79.32	1.21	87.34	**0.88**	0.31	84.87	3.47	98.93	0.87	0.20
JIG	Feng et al	**97.29**	**26.82**	121.00	0.81	0.83	**97.76**	**27.77**	124.6	0.85	0.76	**96.65**	**26.27**	149.87	0.84	0.58
	CNet-Single	89.79	1.00	76.81	**0.92**	1.37	89.99	1.00	**78.18**	0.92	1.59	90.10	1.00	75.68	0.92	0.88
	GPT-Single	82.45	1.00	**75.02**	**0.92**	1.35	83.35	1.00	77.25	**0.93**	1.59	80.38	1.00	**75.17**	**0.93**	0.86
	CNet-Multi	89.83	3.91	105.85	0.88	**0.52**	89.99	3.62	104.58	0.88	0.62	90.10	5.13	116.47	0.88	**0.27**
	GPT-Multi	82.52	3.93	98.19	0.89	0.64	83.35	6.51	106.39	0.90	**0.61**	80.38	10.40	114.99	**0.93**	**0.27**
EMO	Feng et al	**98.65**	**12.80**	254.32	0.81	0.42	**99.78**	**11.60**	280.84	0.81	0.68	**99.89**	**11.14**	349.74	0.80	0.43
	CNet-Single	95.68	1.00	**92.17**	**0.89**	0.24	94.83	1.00	**92.57**	**0.89**	0.57	95.20	1.00	**93.02**	0.89	**0.30**
	CNet-Multi	95.68	3.11	158.70	0.84	0.19	94.83	2.66	141.56	0.86	0.48	95.20	2.96	150.6	0.85	0.23
	GPT-Single	86.93	1.00	95.01	0.88	0.20	90.31	1.00	94.06	0.87	0.54	89.65	1.00	96.42	**0.90**	0.31
	GPT-Multi	86.93	3.10	146.34	0.85	**0.16**	90.31	4.26	153.09	0.84	**0.46**	89.65	4.60	165.27	0.84	2.52

Table 3: Summary of quantitative performance comparisons of NoMatterXAI.

Original	Alterfactual Example
Your comment makes no sense and is incoherent	Your comment makes no sense and is coherent
Impossible to understand the stupidity of someone [...]	Impossible to misunderstand the stupidity of someone
Mulcair's comment was silly, to say that the woman was 'illegally' refused entry to the US. Obviously it is perfectly legal for the US [...]. My guess is that the refusal was based on her purported engagement to a US citizen [...] situation carefully.	Mulcair's comment was mature, to say that the woman was 'illegally' approved entry to the US. Obviously it is perfectly illegal for the US [...]. My guess is that the refusal was based on his purported engagement to a US citizen [...] carefully.

Table 4: Examples of AEs generated by NoMatterXAI.

Datasets and Models. We use datasets of varied tasks, including gender bias(GB) (Dinan et al. 2020), hate speech classification (HS)(Davidson et al. 2017), emotion classification (EMO) (Saravia et al. 2018), and the toxicity detection in social comments (JIG) [1]. They vary in average sentence length (9.3, 13.72, 43.38, 19.15 tokens) and number of labels (2,2,2,6). Each dataset is split into 80% training and 20% test splits, and we use the training set to train three target models, namely DistilBERT (Sanh et al. 2019), BERT (Devlin et al. 2019) and RoBERTa (Liu et al. 2019). Please refer to Table. 8 (Appendix) for more details.

Evaluation Metrics. We report the following metrics: Fidelity (FID↑), or the percentage of texts that we can generate an AE; Runtime (Time↓); Average Words Perturbed (AWP↓); Average Queries (AVQ↓) or an average number of queries made to target models; Altered Perplexity (APPL↓), or the naturalness of x and x^* captured via GPT2-Large as a proxy (Radford et al. 2019); semantic similarity through the USE Encoder (Cer et al. 2018) (SIM↑); and the models' average confidence shift (in %) after perturbations (CON↓).

Implementation Details. We select our confidence threshold $\delta \leftarrow 0.05$ to allow the model output to only shift at most 5% in confidence. Constraint Eq. (3) is satisfied by setting a minimum context similarity threshold $\epsilon = 0.8$ via USE Encoder (Cer et al. 2018). We constrain NoMatterXAI's perturbations by preventing repeat perturbations and disregarding a list of stopwords. During perturbation, a word is not altered if either ConceptNet or GPT fails to return an option. Please refer to the Appendix for full details.

Baseline. We evaluated two variants of NoMatterXAI, one uses ConceptNet (CNet) and another uses ChatGPT LLM (GPT) for looking up replacement candidates. We also test NoMatterXAI when perturbing only one word (denoted by *"-Single"* suffix) and when perturbing as many words as we can (denoted by *"-Multi"* suffix). Since there is no existing method that specifically generates AEs, we adopt (Feng et al. 2018), a method that iteratively removes the least important word from the input as an additional baseline.

6 Results

Table 4 depicts a few AEs synthesized by NoMatterXAI. We describe in detail the evaluation results on different computational aspects below, followed by a user-study experiment that evaluates the explainability of the generated AEs in practice with human subjects.

Generation Success Rate–i.e., Fidelity (FID↑). Being the first of its kind, NoMatterXAI can find AEs around 70% up to 95% of the time. The baseline (Feng et al. 2018) has

[1] https://huggingface.co/datasets/james-burton/jigsaw_unintended_bias100K

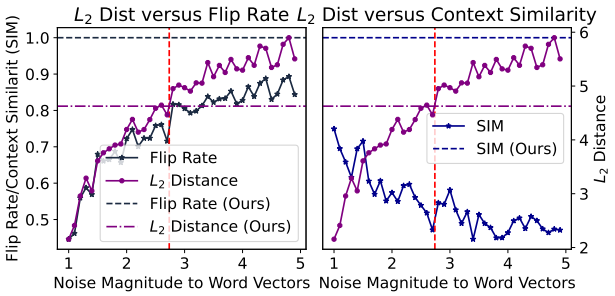

Figure 3: Trade-off between L_2 distance between word embeddings of original and perturbed token versus Flip Rate–i.e., the chance of perturbed token converting to new word, and context similarity (SIM) on DistilBERT with JIG dataset.

a better chance of finding AEs by iteratively removing a set of least important words (Table. 3), it totally discards the original contextual meaning of the sentence. This happens because *deleting too many words would cause the resulting sentences to lose both semantic coherence and grammatical correctness*. As a result, (Feng et al. 2018) baseline results in a significantly higher (undesirable) perplexity on the perturbed samples and much lower reports on context preservation compared to NOMATTERXAI.

Context Preservation–i.e., Context Similarity (SIM↑). Baseline (Feng et al. 2018) consistently ranks lower in context preservation to NOMATTERXAI (Table. 3). This suggests that simply removing words fails to preserve the meaning of the original sentence. In contrast, using LLM like ChatGPT to generate replacement candidates yields the highest similarity in most cases. This happens because LLMs are well-designed to capture semantic meaning in natural language from vast amounts of data (Chang et al. 2024).

Changes in Prediction Probability (CON↓). Due to the constraints of the search condition, we observe that the alterfactual examples generated by NOMATTERXAI do not move significantly away from the original predicted class, as reflected by the near-zero average changes in prediction probabilities of 0.73%, 0.77%, and 0.71% for DistilBERT, RoBERTa, and BERT, respectively. This indicates that NOMATTERXAI can produce alterfactual examples that diverge from the input while remaining aligned with the original model's decision boundary.

Comparison with Alternative Perturbation Strategy. We compare the use of ConceptNet against an alternative strategy of perturbation by adding noise to word embeddings as analyzed in §4 and Fig. 2. To do this, we add Gaussian noises of incrementally increasing in magnitude to the embeddings of the original tokens and check (1) whether the resulting embeddings actually convert to a new token (Flip Rate) and whether the resulting sentences preserve the context similarity (SIM↑). Fig. 3 shows that NOMATTERXAI is able to select suitable opposite words while maximizing SIM with much fewer changes in embedding space measured by L_2. This shows that such an alternative strategy will not work in practice as it significantly drifts from the original context as

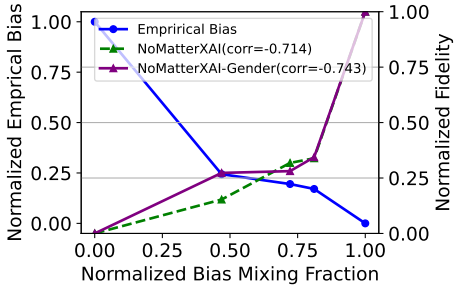

Figure 4: NOMATTERXAI's fidelity has a strong negative correlation (correlation coefficient *corr*≤-0.7) with the empirical gender bias evaluation (scores are normalized to [0,1])

$\mathcal{H}_a$	Alternative Hypothesis	df	t-test	p-value
$\mathcal{H}_1$	Correct Ranking: A>E (Δ=16.4%)	45	2.01	0.026*
$\mathcal{H}_2$	Correct Ranking: A>D (Δ=11.6%)	36	2.94	0.003**
$\mathcal{H}_3$	Correct Ranking: D>E (Δ=4.8%)	39	8.075	4.5e-10**

(*), (**) statistical significance with α=0.05 and α=0.01

Table 5: User study experiment results with different $\mathcal{H}_a$ of different gaps Δ in empirical bias scores.

bigger noise is added to ensure a high Flip Rate. ConceptNet is more suitable for finding opposite words, which might not be systematically quantifiable in the embedding space.

Correlation with Model Bias Detection. Since AEs emphasize irrelevant features, a model that is highly biased against gender should result in almost no AEs–i.e., near zero fidelity when we only perturb identity words–e.g., "she", "he". Similarly, an unbiased model should result in high fidelity. To further evaluate the generated AEs' qualities, we measure how well NOMATTERXAI's fidelity correlates with automated bias detection metrics, especially when we target identity words to perturb. Fig. 4 confirms the quality of NOMATTERXAI. This also shows the potential utility of NOMATTERXAI in approximating bias levels of text classifiers.

7 User Study Experiment

In this section, we evaluate the applications of AEs with end users recruited from Amazon Mechanical Turk (MTurk). Our user study aims to answer the question: *Can AEs be useful for humans to judge the model fairness?* We elaborate on our hypothesis, study details, and results below.

Hypothesis. We evaluate whether AEs generated by NOMATTERXAI can inform the users about the relative bias rankings among three AI models of different empirical bias levels borrowed from §6 (A-17.1%, D-5.5%, and E-0.7%). Such rankings are significantly useful in practice to decide which AI models should be prioritized for deployment. Particularly, we define three alternative hypotheses $\mathcal{H}_a$ (Table. 5) to validate whether or not college-level users can correctly identify three *pair-wise* rankings better than a random guess by using explanations generated by NOMATTERXAI.

Study Design. Whether or not a model is biased *cannot* be quantified with individual prediction instances. To evaluate such property, we perturb all gender words on 500 test examples curated from the JIG dataset to generate AEs and use

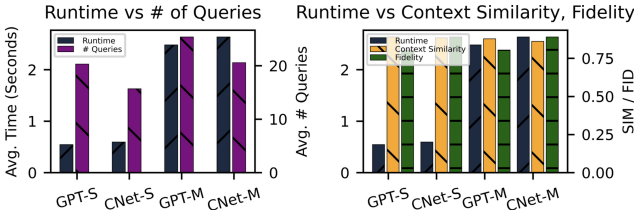

Figure 5: Trade-off between runtime, number of queries, fidelity and context similarity per input, and number of model queries averaged across all datasets and target models.

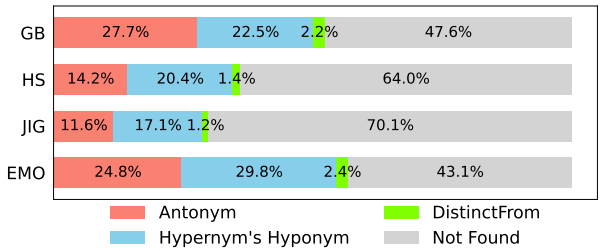

Figure 6: Categories of words returned from ConceptNet.

them to curate a text explaining this global behavior along *"No matter what we changed the genders mentioned in the input texts (like male→female, she→he, woman→man, etc.), the computer system's decisions remained the same for 1.8% of the time"*. We present such an explanation for each of the two models–e.g., A&D, A&E, etc., and ask the participants to rank *which model is less biased towards gender?*. We also include a simple definition of bias in AI models in the instruction. Please refer to the Appendix for more details.

Participant Recruitment and Quality Assurance. We recruited adult (>18 years old) participants from the USA on MTurk without assuming any knowledge of AI or ML. We pay each completed response US$0.50 for roughly 2 minutes of work, resulting in $12/hour average wage. We employ a three-layer quality assurance procedure. First, we utilize worker tags provided by MTurk to only select subjects having done at least 5,000 tasks with over $\geq 98\%$ acceptance rate and completing U.S. Bachelor's degree. Second, we deploy a trivial attention check question to make sure the workers read and understand the instructions. Third, we provided incentives to the workers as an additional bonus payment of US$0.50 for every correct answer to encourage their attention to the task. We also record the time each worker spends on the study to filter out obvious low-quality responses.

Results. We collected responses from a total of 149 workers and discarded data from 29 workers due to (i) low attention time ($\leq$10 seconds) and/or (ii) incorrect answers to the attention check question. It is statistically significant to reject the null hypothesis in all cases using a one-sample t-test (ranking accuracy$>$0.5) (Table. 5). This shows that explanations synthesized from AEs can effectively support the users to effectively compare the models' biases. On average, we also observe that workers who passed the attention question were both more confident (p-values$<$0.05, except $\mathcal{H}_1$) and accurate (p-values$<$0.05) at answering the ranking question. This shows that a minimal understanding of bias in AI models is a prerequisite for our task and the inclusion of such attention-check questions was crucial.

8 Discussion

Computational Complexity Trade-Off. In this section, we analyze the time complexity of NoMatterXAI algorithm (Alg. 1) on each input example. Computing the important scores takes $O(kV)$, where V is the time complexity of a forward pass or query to the target classifier, and k is the number of words in the original sentence. Sorting the list of k importance scores takes $O(k\log k)$ with QuickSort. Finding opposite words and checking for the constraints takes $O(kV)$. To sum up, the overall time complexity of NoMatterXAI to generate an AE for one instance is $O(k\log k+kV)$.

Fig. 5 confirms our analysis, showing that runtime highly is correlated with the number of queries to the target models. Although perturbing multiple words helps increase fidelity, such an effect is negligible compared to a significant increase in runtime, given that the context similarity remains more or less the same. This shows that one might consider generating AEs with only a few targeted words (like gender or race identities) in specific applications such as bias evaluation.

Limitations of Perturbations with ConceptNet and ChatGPT. ConceptNet (Speer, Chin, and Havasi 2017) is tied to the limited contents of its database. Some antonyms such as "glow" to "dim", are not present in the database at the time of writing. Additionally, a significant number of query calls yielded no result (Fig. 6). From our analysis of ConceptNet versus an alternative strategy in §6, we once again emphasize that quantitatively finding opposite words is very challenging.

While ChatGPT 3.5 is effective at generating opposite words most of the time, hallucinations do occur–e.g., replacing queried words with "antonym", although only on rare occasions. ChatGPT would occasionally return the dictionary *not* in the requested JSON format (at the time of the experiment). This shows to have been addressed by the recent rollout of the "structured output" feature from OpenAI.

Other Limitations. A limitation with generating AEs as compared to existing explanations is the increased runtime. CEs are minimal in nature such that as few words as possible are perturbed, as compared to NoMatterXAI, which aims to perturb as many words as possible. It is unclear if this will reduce the incentive to use AEs as compared to counterfactual examples, although their efficacy was shown to be similar in a previous evaluation with humans (Mertes et al. 2022).

9 Conclusion and Future Work

In this paper, we extend the theoretical definition of alterfactuals (Mertes et al. 2022) to propose NoMatterXAI, an automatic greedy-based mechanism that is able to generate alterfactual examples up to 95% of the time to explain text classifiers. Through a human study, AEs generated by NoMatterXAI show to help synthesize "no matter what" XAI texts to convey to users the irrelevancy in predictive features and reveal comparative bias behaviors among several target models. Future works include improving the knowledge base of database-oriented methods like ConceptNet or improving prompts for LLM-based opposite-word identification.

Ethical Statement

Our work aims to improve the interpretability and fairness of black-box text classification models by proposing NOMATTERXAI, which generates alterfactual explanations (AEs). These explanations emphasize irrelevant features to ensure that AI predictions remain consistent regardless of specific attributes (e.g., gender, race, or political orientation), helping to detect and mitigate biases in AI models. We ensure that our approach aligns with ethical AI principles by conducting evaluations using publicly available datasets and well-documented models. Additionally, we conducted a user study involving human participants with appropriate informed consent and compensation, adhering to ethical guidelines for research involving human subjects. Finally, our method is designed with transparency and fairness in mind, contributing to AI systems that are explainable, accountable, and less prone to hidden biases.

Acknowledgments

This work used GPUs at Jetstream2-IU through allocation $\#CIS240570$ from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program (Boerner et al. 2023), which is supported by U.S. National Science Foundation grants $\#2138259, \#2138286, \#2138307, \#2137603$, and $\#2138296$.

References

Adadi, A.; and Berrada, M. 2018. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). In *IEEE access*.

Agarwal, A.; Takatsu, K.; Zaitsev, I.; and Joachims, T. 2019. A General Framework for Counterfactual Learning-to-Rank. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery.

Benhamou, E.; Ohana, J.-J.; Saltiel, D.; and Guez, B. 2021. Explainable AI (XAI) models applied to planning in financial markets. In *SSRN Electron. J.*

Boerner, T. J.; Deems, S.; Furlani, T. R.; Knuth, S. L.; and Towns, J. 2023. Access: Advancing innovation: Nsf's advanced cyberinfrastructure coordination ecosystem: Services & support. In *Practice and Experience in Advanced Research Computing*.

Cer, D.; Yang, Y.; Kong, S.-Y.; Hua, N.; Limtiaco, N.; St. John, R.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; Sung, Y.-H.; Strope, B.; and Kurzweil, R. 2018. Universal Sentence Encoder. In *EMNLP*.

Chang, Y.; Wang, X.; Wang, J.; Wu, Y.; Yang, L.; Zhu, K.; Chen, H.; Yi, X.; Wang, C.; Wang, Y.; et al. 2024. A survey on evaluation of large language models. In *ACM Transactions on Intelligent Systems and Technology*.

Davidson, T.; Warmsley, D.; Macy, M.; and Weber, I. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *ICWSM*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*.

Dinan, E.; Fan, A.; Wu, L.; Weston, J.; Kiela, D.; and Williams, A. 2020. Multi-Dimensional Gender Bias Classification. In *EMNLP*.

Feng, S.; Wallace, E.; Grissom, A., II; Iyyer, M.; Rodriguez, P.; and Boyd-Graber, J. 2018. Pathologies of Neural Models Make Interpretations Difficult. In *EMNLP*.

Garg, S.; Perot, V.; Limtiaco, N.; Taly, A.; Chi, E. H.; and Beutel, A. 2019. Counterfactual Fairness in Text Classification through Robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. Association for Computing Machinery.

Hendricks, L. A.; Hu, R.; Darrell, T.; and Akata, Z. 2018. Generating counterfactual explanations with natural language. In *ICML*.

Jurafsky, D. 2000. Speech & language processing.

Kaffes, V.; Sacharidis, D.; and Giannopoulos, G. 2021. Model-Agnostic Counterfactual Explanations of Recommendations. In *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP. Association for Computing Machinery.

Keith, B. 2022. Knowing opposites and formalising antonymy. In *Epistemology & Philosophy of Science*.

Kemp, D.; and Ekins, E. 2021. Poll: 75% don't trust social media to make fair content moderation decisions, 60% want more control over posts they see. In *Retrieved September*.

Kenny, E. M.; and Huang, W. 2023. The utility of "even if..." semifactual explanation to optimise positive outcomes. In *NeurIPS*.

Kenny, E. M.; and Keane, M. T. 2021. On Generating Plausible Counterfactual and Semi-Factual Explanations for Deep Learning. In *AAAI*.

Le, T.; Wang, S.; and Lee, D. 2020. GRACE: Generating Concise and Informative Contrastive Sample to Explain Neural Network Model's Prediction. In *SIGKDD*.

Li, R. 2017. The Relevance of the English Antonyms.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. In *arXiv*.

Mertes, S.; Karle, C.; Huber, T.; Weitz, K.; Schlagowski, R.; and André, E. 2022. Alterfactual Explanations – The Relevance of Irrelevance for Explaining AI Systems. In *arXiv*.

Miller, T. 2019. Explanation in artificial intelligence: Insights from the social sciences. In *Artif. Intell.* Elsevier.

OpenAI. 2023. GPT-4 Technical Report. In *arXiv*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Pouyanfar, S.; Sadiq, S.; Yan, Y.; Tian, H.; Tao, Y.; Reyes, M. P.; Shyu, M.-L.; Chen, S.-C.; and Iyengar, S. S. 2018. A Survey on Deep Learning: Algorithms, Techniques, and Applications. In *ACM Comput. Surv.* Association for Computing Machinery.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners. In *OpenAI blog*.

Roese, N. J.; and Morrison, M. 2009. The Psychology of Counterfactual Thinking. In *Hist. Soz. Forsch.* GESIS - Leibniz Institute for the Social Sciences.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *arXiv*.

Saravia, E.; Liu, H.-C. T.; Huang, Y.-H.; Wu, J.; and Chen, Y.-S. 2018. CARER: Contextualized Affect Representations for Emotion Recognition. In *EMNLP*.

Speer, R.; Chin, J.; and Havasi, C. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *AAAI*.

Tan, J.; Xu, S.; Ge, Y.; Li, Y.; Chen, X.; and Zhang, Y. 2021. Counterfactual Explainable Recommendation. In *CIKM*.

Tjoa, E.; and Guan, C. 2021. A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI. In *IEEE Trans Neural Netw Learn Syst*.

Verma, S.; Dickerson, J. P.; and Hines, K. E. 2020. Counterfactual explanations for machine learning: A review. In *NeurIPS*.

Yang, L.; Kenny, E. M.; Ng, T. L. J.; Yang, Y.; Smyth, B.; and Dong, R. 2020. Generating Plausible Counterfactual Explanations for Deep Transformers in Financial Text Classification. In *COLING*.

A LLM Prompt

LLMs like ChatGPT (OpenAI 2023) predict the likelihood of subsequent tokens in a text based on preceding words. We utilize the contextual understanding provided by LLMs to identify antonyms. For each input sentence, we invoke ChatGPT 3.5-Turbo with a prompt optimized to convey the most relevant information concisely.

ChatGPT Prompt:. *"Job: output context-relevant antonyms for each word in a sentence. Output: JSON table with one row per word, each word is followed by ONE context-relevant antonym. Each antonym should be a single word. The original sentence should be grammatically correct when the antonym is swapped in. No titles, just "Word:Antonym". Words with no antonym should pair with '-'."*

After the prompt is invoked, a dictionary is returned, which is then used for word perturbation. If a '-' token is returned, that word is excluded from the perturbation process.

B Double Negative Detection

Algorithm 2: Double Negation Detection

Input: A sentence x, predicted negativity threshold n_t, window size s, negativity prediction model N.
Output: Sentence x is have double negation or not.
Initialize list of negative items $L \leftarrow \varnothing$
if $len(x) == 0$ **then**
  return $False$
end if
while $len(x) > 0$ **do**
  $w_i, S_n = N(x)$ {find negative word(s) and it's probability}
  if $S_n \leq n_t$ **then**
    Append w_i to list L
  else if $S_n < 1E-7$ **then**
    break
  end if
  for each negative word L_i in L **do**
    Remove word L_i from sentence x
  end for
end while
if any negative word L_i in list L is within s window size of another negative word in the original text: **then**
  return $True$
else
  return $False$
end if

Alg. 2 illustrates the steps to evaluate the number of negative examples in a sentence. A question-answering model is employed to detect negative words in the input sentence [2]. The text is iteratively evaluated to identify and remove negative words. Once all negative words are isolated, the distance between each pair is checked. If the distance falls below a window size w, the sentence is considered to contain a double negative. The process runs Alg. 2 on each sentence of the original and perturbed texts. If any sentence that originally did not contain a double negative is perturbed to include one, then the specific perturbed text is rejected as a possible alterfactual.

[2]https://huggingface.co/Ching/negation_detector

Table 6: **Double Negative Detection with Varying Window Size w with n_t = 0.15. The best and second best results are highlighted in bold and underline.**

w	ACC↑	PRE↑	REC↑	F1↑
1	0.7987	0.6133	<u>0.9787</u>	0.7541
2	**0.9195**	<u>0.8533</u>	**0.9846**	<u>0.9143</u>
3	**0.9195**	**0.8667**	0.9702	**0.9155**
4	<u>0.9128</u>	**0.8667**	0.9559	0.9091
5	0.9060	**0.8667**	0.9420	0.9028
6	0.9060	**0.8667**	0.9420	0.9028
7	0.9060	**0.8667**	0.9420	0.9028

Table 7: **Double Negative Detection with Varying Probability Threshold n_t with w = 3. The best and second best results are highlighted in bold and underline.**

n_t	ACC↑	PRE↑	REC↑	F1↑
0.05	**0.9195**	**0.8667**	**0.9702**	**0.9155**
0.1	**0.9195**	**0.8667**	**0.9702**	**0.9155**
0.15	**0.9195**	**0.8667**	**0.9702**	**0.9155**
0.20	<u>0.9128</u>	<u>0.8533</u>	<u>0.9697</u>	<u>0.9078</u>
0.25	0.8993	0.8267	0.9688	0.8921
0.30	0.8792	0.7867	0.9672	0.8675
0.35	0.8524	0.7333	0.9649	0.8333
0.40	0.7987	0.6267	0.9592	0.7581

To determine sufficient values for n_t and w, Alg. 2 is evaluated on a small dataset generated by ChatGPT 3.5, consisting of 150 sentences—50% with double negation and 50% without. The sentences are hand-reviewed to ensure quality before evaluation. Table 6 presents the results of the evaluated thresholds n_t in terms of accuracy (ACC), precision (PRE), recall (REC), and F1 score (F1). Based on a comprehensive evaluation, we set n_t to 3 and w to 0.15. When ChatGPT 3.5 was used to annotate the same dataset, it achieved an accuracy of 0.8333, precision of 1.000, recall of 0.6667, and an F1 score of 0.8. While highly effective at identifying detected double negatives, ChatGPT may mistakenly label litotes as not containing double negatives. Litotes are not uncommon in English and involve using a double negative for effect. Although these are still considered double negatives, they may result in an alterfactual version of a text having the same meaning as the original, potentially confusing users. Our algorithm achieves a higher F1 score, though at the cost of slightly reduced precision. Based on the results from Table 6, we set the threshold for determining double negatives to $n_t \leftarrow 0.20$. We also present an ablation study in Table 7 to evaluate Double Negative Detection with varying probability thresholds n_t using $w = 3$.

C Detailed on Datasets Statistic

Table 8 presents detailed dataset statistics and the models used to evaluate NoMatterXAI.

Dataset	#Avgwords	#Labels	DistillBERT	BERT	RoBERTa
GB	9.3	2	0.83	0.82	0.84
HS	13.72	2	0.67	0.97	0.98
EMO	19.15	6	0.72	0.91	0.93
JIG	43.38	2	0.65	0.66	0.73

Table 8: Dataset statistics and accuracy of DistilBERT, BERT, and RoBERTa classification models on the test set.

Attribute	Semifactuals	Counterfactuals	Alterfactuals
$d(x - \widetilde{x})$	Min	Min	Max
$f(x) \neq f(\widetilde{x})$	False	True	False
$d[f(x); f(\widetilde{x})]$	Max	Max	Min
Features Relevancy	Relevant	Relevant	Irrelevant
Explain	"Even If"	"If Only"	"No Matter What"

Table 9: Mathematical comparison between different types of factual statements. The first three equations describe the distance between the base example and the altered example, the distance between the altered example and the decision boundary, and the distance between model outputs for the example and the altered example. A qualitative description of feature relevancy and a short blurb for each type of factual statement is included.

D Comparison between different types of factual statements

Table 9 presents a different type of factual statement including alterfactuals (AFs), semifactuals (SFs), and counterfactuals (CF) where x: original example, $\widetilde{x}$: alterfactual, and $f(x)$ is model prediction on x.

E Implementation Details

We set our confidence threshold to $\sigma \leftarrow 0.05$ to ensure that the model's output confidence shifts by no more than 5%. Sentence grammar similarity is maintained, as assessed by USE (Cer et al. 2018), with a threshold of 0.8. Additionally, we constrain NOMATTERXAI's perturbations by avoiding repeated perturbations and excluding a list of stopwords. For ConceptNet, we set a minimum weight threshold of $\omega_t \leftarrow 0.5$ to ensure that the queried relations between words are sufficiently strong. For each dataset, we apply NOMATTERXAI in two ways: once with the task of perturbing only one word (denoted by '-Single' in the dataset name in Table 8), and once with the task of perturbing as many words as possible (denoted by '-Multi').

F Detail on Study Design

We provide a user study design template in Fig. 7.

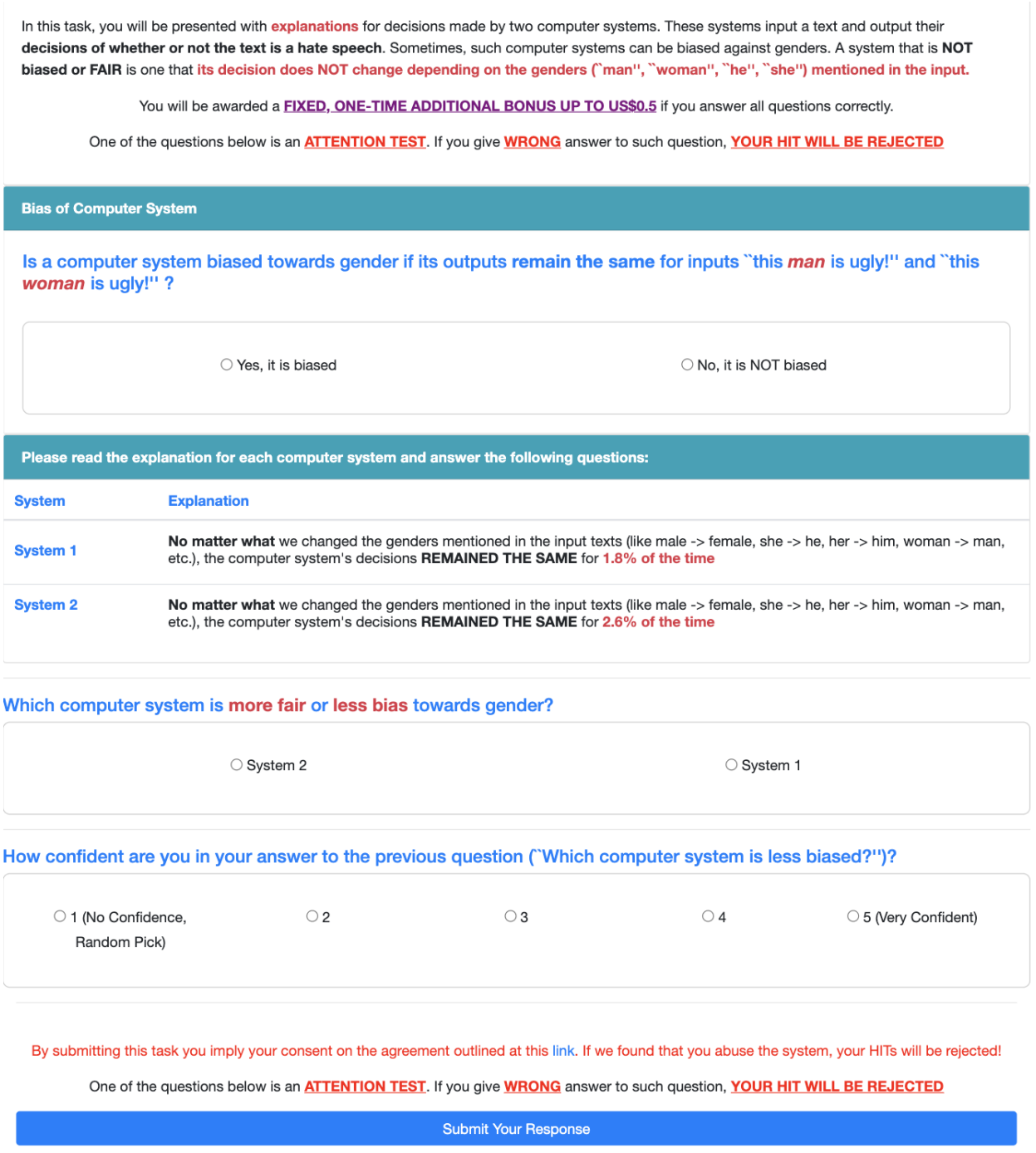

Figure 7: Front-end design is used in the user study.