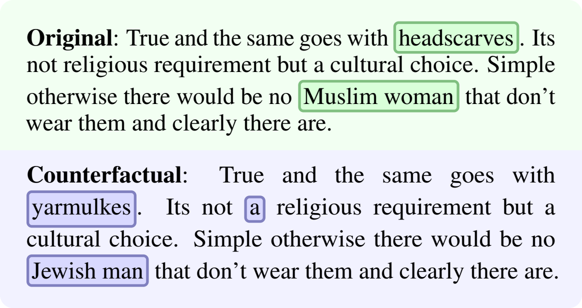
Flexible text generation for counterfactual fairness probing

Zee Fryer[*] **Vera Axelrod** **Ben Packer** **Alex Beutel** **Jilin Chen** **Kellie Webster**
Google Research
{zeef, vaxelrod, bpacker, alexbeutel, jilinc, websterk}@google.com

Abstract

A common approach for testing fairness issues in text-based classifiers is through the use of counterfactuals: does the classifier output change if a sensitive attribute in the input is changed? Existing counterfactual generation methods typically rely on wordlists or templates, producing simple counterfactuals that don't take into account grammar, context, or subtle sensitive attribute references, and could miss issues that the wordlist creators had not considered. In this paper, we introduce a task for generating counterfactuals that overcomes these shortcomings, and demonstrate how large language models (LLMs) can be leveraged to make progress on this task. We show that this LLM-based method can produce complex counterfactuals that existing methods cannot, comparing the performance of various counterfactual generation methods on the Civil Comments dataset and showing their value in evaluating a toxicity classifier.



Original: True and the same goes with headscarves. Its not religious requirement but a cultural choice. Simple otherwise there would be no Muslim woman that don't wear them and clearly there are.

Counterfactual: True and the same goes with yarmulkes. Its not a religious requirement but a cultural choice. Simple otherwise there would be no Jewish man that don't wear them and clearly there are.

Figure 1: Counterfactual generated by our LLM-based method, given the original text and the prompt "make this not about Muslims".

1 Introduction

It is well known that classifiers (such as toxicity detectors) can pick up negative associations about marginalized groups from their training data, e.g. due to under-representation of those groups in the training data, or the higher levels of toxicity in the text data referring to these groups (Sap et al., 2019; Dixon et al., 2018; Zhou et al., 2021).

One common method of testing classifier models for these unwanted associations is by comparing the classifier's outputs on a particular type of counterfactual text pair: specifically, text pairs which are as similar as possible in format and meaning, but such that one text references a particular sensitive attribute and the other does not (Figure 1; here the sensitive attribute is Islam). If the classifier exhibits a large number of "flips" (changes in prediction from original to counterfactual) on these pairs, this indicates a potential problem that may be addressed through mitigations such as dataset augmentation (Dixon et al., 2018) or counterfactual logit pairing (Garg et al., 2019).

Here we focus on counterfactual generation, and specifically the following questions: 1) How can we efficiently generate large datasets of counterfactual pairs? 2) While preserving the diversity, fluency and complexity of real-world inputs? 3) To probe for subtle or previously-unknown issues?

One approach is to ask humans to create counterfactual counterparts by editing existing examples, but this can be both costly and slow (see e.g. §3 in Kaushik et al. (2020)). Another method is to use human-curated wordlists to generate counterfactuals: for example to apply ablation or substitution on existing texts or to fill in preset templates (Garg et al., 2019; Dixon et al., 2018; Rudinger et al., 2018; Sheng et al., 2019a). While these approaches can efficiently generate large datasets (excluding the time required to create the initial wordlists), the results often fail to be fluent, diverse or complex (as we show in Section 5) and are not likely to uncover novel issues that the wordlist creators had not considered.

We suspect that as it becomes more common to use large language models (LLMs) as the base for

[*] Work done as a Google AI Resident.

Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH), pages 209 - 229
July 14, 2022 ©2022 Association for Computational Linguistics

classifier models (such as toxicity classifiers), these classifiers will become more sensitive to factors such as fluency, word order, and context, and counterfactual generation methods will need to evolve correspondingly to keep up.

With this in mind, we define a new counterfactual generation task (Section 3.1) and demonstrate the potential of existing LLM techniques to address this problem (Section 3.2). Specifically, we show how ideas from Reif et al. (2021) can be used to generate natural, diverse, and complex counterfactuals from real-world text examples (as in Figure 1) and combine this with both automated and human evaluation methods (Section 3.3) to ensure that the resulting counterfactuals are of high quality and suited to the task at hand. This human-in-the-loop component also helps to mitigate the risks introduced by using an LLM to generate the text (Section 3.4). Finally, we compare the performance of our method with existing counterfactual generation methods (Section 5), and show that existing methods may not capture certain subtle issues in toxicity classifiers, and that our method addresses some of these deficiencies (Section 5.3).

We use toxicity detection as a testbed in this work, and focus on generating counterfactuals to probe for false positives – that is, non-toxic text which is misclassified as toxic due to identity references. While we focus on this particular application to demonstrate one way in which our framework can be useful, it could also be applied in other contexts: for example, probing for false negatives, applications other than toxicity detection, and counterfactual perturbations other than removing the presence of a sensitive attribute.

2 Related Work

2.1 Counterfactual generation

Two common types of counterfactual text pair generation are 1) rule-based methods using templates and/or wordlists, and 2) controlled text generation using deep learning-based language models.

Template-based counterfactual datasets are often built from short, simple sentences: for example, the Jigsaw Sentence Templates dataset consists of templates such as "I am a ⟨adjective⟩ ⟨identity-label⟩" and "I hate ⟨identity-label⟩".[1] Other examples include Rudinger et al. (2018); Sheng et al. (2019a). While this approach provides fine-grained control over identity references and toxicity balance, it also has disadvantages: for example, the resulting text is often not natural and looks quite different from the actual task data. Works such as Prabhakaran et al. (2019) and Hutchinson et al. (2020) partially mitigate this by using real-world data and targeting specific syntactic slots for substitution, but this can yield incoherent or contradictory text when there are multiple entities referenced in a sentence. Finally, recent works with templates such as Röttger et al. (2021) and Kirk et al. (2021) have been effective at detailing problems with modern toxicity classifiers, by investing significant targeted effort into probing task-specific functionality, and employing human validation for generated examples.

There have also been attempts to use deep learning to build more general-purpose counterfactual generators. One example is Polyjuice (Wu et al., 2021), which combines a finetuned GPT-2 model with control codes to generate diverse natural perturbations. But, as we show below, it is difficult to use Polyjuice to modify references to a pre-specified topic. Another example is CAT-Gen (Wang et al., 2020), which trains an RNN-based encoder-decoder model, using a separate attribute classifier to guide the decoder towards the modifying the desired attribute. However, both of these require large training sets labeled by sentence permutation type (Polyjuice) or attribute (CAT-Gen).

Other methods combine a pretrained language model with a task-specific classifier, e.g. Dathathri et al. (2020) and Madaan et al. (2020) which both leverage Bayes' rule to guide text generation while avoiding the need to retrain or finetune the language model itself, and Davani et al. (2021) which uses GPT-2 to generate text and uses wordlists and likelihood thresholds to identify valid counterfactuals. ToxiGen (Hartvigsen et al., 2022) uses GPT-3 with and without an adversarial classifier-in-the-loop method to generate a large set of challenging examples for toxicity detection, employing identity-specific engineered prompts. Our method is most similar to these approaches, though we rely less on task-specific classifiers and use generic prompts.

2.2 Counterfactual evaluation

While most counterfactual generation work includes a definition of what constitutes a "good" counterfactual and some method of measuring success relative to these desiderata, the definitions and methods vary depending on factors such as the intended downstream use of the counterfactuals.

[1] https://github.com/conversationai/unintended-ml-bias-analysis

Attribute	Original	LLM-D **rewrite**
LGBQ+	How is "embracing and accepting" their *homosexuality* not a lifestyle choice?	How is "embracing and accepting" their **love** not a lifestyle choice?
transgender	Some people are born *transgender*. That appears to be a verifiable fact. Why is this a question of "left" or "right"?	Some people are born **left-handed**. That appears to be a verifiable fact. Why is this a question of "left" or "right"?
Judaism	Get *JPFO* up here. If anyone has anything to say about guns it is that organization. For those that do not know. *JPFO is Jews for the Preservation of Firearms*.	Get **the NRA** up here. If anyone has anything to say about guns it is that organization. For those that do not know. **NRA is for National Rifle Association**.
Islam	If its *Muslim* he's all over it.... I can't figure this guy's loyalty. Who is influencing this guy..... Is it the *Muslim Brotherhood, Saudi Arabia, Qatar*??	If it's **American** he's all over it.... I can't figure this guy's loyalty. Who is influencing this guy..... Is it the **Democrats, Republicans, Supreme court**??

Table 1: Examples of LLM-D-generated counterfactuals, demonstrating LLM-D's ability to make neutral context-aware substitutions or multiple consistent substitutions to remove explicit and implicit references.

Many methods prize counterfactuals with minimal edits relative to the original text and measure success using distance, e.g. Ross et al. (2021a) Madaan et al. (2020). However, this is not well suited for evaluating counterfactuals generated from longer or complex original texts, as these often require multiple edits to remove all references to the sensitive attribute. Some methods reward grammaticality but do not require the text to make semantic sense (Sheng et al., 2020), while others require both fluency and consistency (Ross et al., 2021b; Reif et al., 2021; Madaan et al., 2020); some use automated metrics such as perplexity (Wang et al., 2020) and masked language model loss (Ross et al., 2021a) while others use human raters to evaluate fluency (Reif et al., 2021; Wu et al., 2021).

Building on these prior results, we combine several automated metrics to filter out poor quality counterfactuals (e.g. ones with large additions/deletions beyond those required to remove the sensitive attribute). We also develop a human evaluation framework to rate the quality of the counterfactuals that pass automated filtering, with a view to making it easy for human annotators to rate examples quickly and consistently while also rewarding diverse and non-obvious counterfactual generation (e.g. rows 1 and 2 of Table 1).

2.3 Toxicity detection

It is well documented (Davidson et al., 2019; Dixon et al., 2018) that toxicity and hate speech classifiers often pick up on correlations (that are not causations) between references to certain identities and toxic speech: that is, these models incorrectly learn that sensitive attributes such as certain sexual orientations, gender identities, races, religions, etc. are themselves indications of toxicity.

Recent work has gone further and explored the effect of *indirect* toxic examples on classifiers (Sap et al., 2020; Lees et al., 2021; Han and Tsvetkov, 2020), finding that many datasets do not adequately represent this form of toxicity (Breitfeller et al., 2019) and that classifiers are ineffective at identifying it (Han and Tsvetkov, 2020). Based on this, we conjecture that toxicity classifiers may also associate *indirect* references to sensitive attributes with toxicity, which is consistent with (Hartvigsen et al., 2022). We focus on exploring this facet of counterfactual probing.

3 Methodology

Our goal is to detect when a model produces a different score for two examples (original and counterfactual) that differ only by changing a sensitive attribute and that have the same groundtruth label. Ideally the dataset of counterfactual pairs used in this testing should be both large in size and diverse in topic in order to maximise the chances of identifying issues with the model, including issues that the dataset creators may not have considered.

3.1 Task Definition

We define our task as follows:

Given a corpus of text examples that reference a specific sensitive attribute (e.g. a particular religion, LGBQ+ identity, transgender identity), generate a counterfactual text for each original text that preserves both the original label and the original meaning (as far as possible) while removing all references to the chosen sensitive attribute.

Taken as a set, the counterfactuals should be:

- **Complex***: The texts should reflect the complexity of expected real-world inputs.*
- **Diverse***: The counterfactual edits should*

cover a range of topics, both within the attribute's category (e.g. replacing one religion with another) and more generally (replacing specific references with neutral words such as "person", "religion", etc).

- **Fluent and consistent***: The generated text should match the style and phrasing of the input text, should be internally consistent (e.g. no changing topic part way through), and should read like plausible natural language.*

3.2 Counterfactual Generation with LLMs

To generate our counterfactuals we build on the results of Reif et al. (2021), which accomplishes a wide range of style transfers using a Transformer-based large language model combined with prompting. Inputs to the LLM consist of three parts: a small fixed set of prompts that demonstrate the rewriting task, the piece of text to be rewritten, and an instruction such as "make this more descriptive" or "make this include a metaphor". The LLM returns up to 16 attempts at rewriting the input text according to the given instruction.

In order to use this method for counterfactual generation, we retain the prompts used in Reif et al. (2021) (see Table 8 for the full prompt text) but replace the style transfer instruction with ones specific to our task, e.g. "make this not about Muslims" or "make this not about transgender people" (see Appendix A.2 for details). This is one of the few parts of our pipeline that uses the sensitive attribute, and this generalises easily to other attributes simply by changing the instruction.

We use LaMDA (Thoppilan et al., 2022) as the underlying LLM for text generation in this paper, which belongs to the family of decoder-only Transformer-based dialog models. The LaMDA model used here, which we refer to as LLM-D, is described in §6 of Thoppilan et al. (2022): it has 137B parameters, and was pretrained as a general language model (GLM) on 1.97B public web documents and finetuned into a dialog model on an additional dataset of curated dialog examples.

For the experiments reported here, we exclusively used the finetuned dialog model: both for safety reasons (LLM-D's finetuning includes a focus on reducing toxic text generation (Thoppilan et al., 2022)) and technical reasons (it could generate longer passages of text than other models available to us). However, we also achieved success using our method with the underlying GLM model (referred to as "PT" in Thoppilan et al. (2022)), and since prompting techniques have achieved success on multiple different language models (Reif et al., 2021; Brown et al., 2020) we expect that our method would generalise to other LLMs.

3.3 Counterfactual Evaluation

We evaluate counterfactuals in two phases: an automated phase using a combination of standard metrics and a simple two-layer classifier, and a human evaluation phase based on criteria we developed for rating complex counterfactuals. A key consideration here is that while counterfactuals should be as similar as possible to the originals, they must also remove sensitive attribute references; thus we cannot be *too* strict in enforcing similarity, especially via automated methods.

LLM-D was configured to generate up to 16 responses for each input, so we use a combination of automated metrics to identify potential good counterfactuals to pass to human raters. In addition to some simple filtering rules (e.g. to catch examples where LLM-D simply regurgitates its prompt) we use three main metrics:

- BLEU score (Papineni et al., 2002),
- BERTScore (Zhang et al., 2020), and
- a prediction of whether the sensitive attribute is still referenced (described below).

A high BLEU score relative to the original text indicates high lexical similarity (Reif et al., 2021, Appendix B), while a high BERTScore indicates semantic similarity; based on early (separate) tuning experiments, we found that requiring both scores to be above 0.5 was a good trade-off between producing plausible counterfactuals while also allowing some diversity of responses.

The sensitive attribute predictor is a two-layer fully connected classifier trained for this purpose; full training details are given in Appendix A.3. We imposed a threshold of 0.5 on this classifier as well, although the results in this paper suggest that this would benefit from further tuning.

Our human evaluation criteria evaluate the (original, counterfactual) pair along four axes:

1. fluency/consistency,
2. presence of sensitive attribute,
3. similarity of label, and
4. similarity of meaning.

Raters are asked whether the proposed counterfactual is fluent and consistent (yes/no/unsure), whether it references the sensitive attribute (explicitly/implicitly/not at all), whether it should be assigned the same label as the original (yes/no/unsure),[2] and whether it is similar in meaning and format to the original (scale of 0 to 4). The full rater instructions are given in Appendix C.

We use majority vote to consolidate annotator ratings for each example, discarding ties. For our purposes, a counterfactual is deemed "good" if it is fluent, does not reference the sensitive attribute, has the same label as the original, and scores at least 2 (out of 4) on similarity of meaning. Thus examples where the majority vote resulted in a rating of "unsure" were treated as if they had been rated "no" when reporting results in Section 5.

Quantifying "similarity of meaning" "Similarity of meaning" was the hardest metric to define, since removing references to the sensitive attribute often required major edits to the input text. Thus, our score buckets split the counterfactuals in a way that captures both type and severity of edit. This allows us to identify a more diverse pool of good counterfactuals, while also making it easy for users to select a stricter subset if required.

A score of 4 indicates a perfect ablation counterfactual with no unnecessary changes or new information, 3 means that the counterfactual contains substitutions to similar or neutral words (e.g. "Muslim" → "Christian", "Judaism" → "religion"; useful for comparing classifier predictions among identities within a category), while 2 allows for more diverse edits such as minor additions/deletions or substitutions to other topics (useful for initial fairness probing of a model). 1 indicates an example that is reasonably similar to the original but too different to be a useful counterfactual, and a 0 indicates that the text is changed beyond recognition. See Appendix C for full guidelines and examples.

3.4 Safety

Large language models come with safety and toxicity issues (Bender et al., 2021; Abid et al., 2021), which is of particular concern when using them to generate text for the purpose of counterfactual fairness probing in other models. The LLM-D model has been finetuned by its creators to help mitigate some of these safety concerns (Thoppilan et al., 2022, §6), and we also built safeguards into our pipeline to reduce the chances of our method producing problematic or toxic counterfactuals. Even with human-in-the-loop, it is still possible for our method to produce some problematic examples, e.g. ones that perpetuate negative stereotypes, but we aim to reduce this risk.

First, we only aim to generate counterfactuals in the sensitive → neutral direction. That is, we choose input texts that reference the sensitive attribute, and ask LLM-D to remove these references; we do NOT ask the model to generate text about marginalized groups starting from neutral texts (though in practice it can sometimes substitute one identity group for another). Additionally, our evaluation setup ensures that all generated text is checked by at least one human, specifically includes a criteria checking for *implicit* references to the identity as well as explicit ones, and includes a "reject for other reason" box to allow raters to remove examples if either the original or counterfactual text contains negative stereotypes or hate speech. This provides a second line of defence against any toxic text that might slip through.

4 Implementation

4.1 Data

The CivilComments dataset (Borkan et al., 2019) is a set of approximately 2 million English-language internet comments from 2015-2017, with crowdsourced toxicity annotations. CivilComments-Identities (CC-I) is a 450k subset of CivilComments where each text has additional crowdsourced labels for references to various identities, such as gender,[3] sexual orientation, religion, and race.

Our experiments focus on four identity subcategories in CC-I, namely `muslim`, `jewish`, `transgender`, and `homosexual_gay_and_lesbian`,[4] which for simplicity we refer to as `LGBQ+`. These categories were chosen because they are all groups that have faced high levels of online toxicity that may have bled through into classifier models (e.g. Abid et al. (2021); Borkan et al. (2019)), and because the annotators we used for our experiments were sufficiently familiar with these categories to evaluate our generated counterfactuals effectively.

[2] Note that this criteria is task-dependent; in our case the labels were toxic/nontoxic.

[3] The available labels in this category are `male`, `female`, `transgender`, and `other_gender`, which should always be used with extreme care to avoid any implication that "male"/"female" and "transgender" refer to disjoint sets (see e.g. Larson (2017)); for this work we use only the `transgender` label.

[4] Despite the name, this category appears to have been more broadly interpreted by crowd raters as referring to all non-straight sexual orientations, and therefore we continued to treat it as such in our experiments.

Method	Text	F	A	L	M
Original	*So you are saying it's OK? Not enough? Long way to go? An apologist for Islamic terrorism?*	-	-	-	-
Ablation	so you are saying it's ok? not enough? long way to go? an apologist for **terrorism**?	✓	✓	✓	4
Substitution	so you are saying it's ok? not enough? long way to go? an apologist for **christian** terrorism?	✓	✓	✓	3
Polyjuice	So you are saying it's OK? Not enough? Long way to go? An apologist for **secular enlightenment**?	✓	✓	✓	2
LLM-D	So you are saying it's OK? Not enough? Long way to go? An apologist for **terrorism**?	✓	✓	✓	4
Original	*Shocking that this article didn't once mention "islam", "islamic" or "Muslim".*	-	-	-	-
Ablation	shocking that this article didn't once mention **""**, **""** or **""**.	✗	✓	-	-
Substitution	shocking that this article didn't once mention "**christianity**", "**christian**" or "**christian**".	✓	✓	✓	3
Polyjuice	**Waving** that this article didn't once mention "islam", "islamic" or "**islandic**".	✗	✗	-	-
LLM-D	Shocking that this article didn't once mention "**christian**", "**christians**" or "**Christ**".	✓	✓	✓	3

Table 2: Civil Comments examples referencing Islam, with generated counterfactuals and human annotations (**F**luent, doesn't reference **A**ttribute, similarity of **L**abel, and similarity of **M**eaning). In the top example all methods produced a reasonable counterfactual; in the second, only LLM-D and substitution generated good counterfactuals.

We restrict our dataset to texts between 10 and 45 words long that do not contain URLs, for ease of analysis by human raters. We further require that texts have a score of at least 0.8 for the relevant attribute, and a toxicity score of at most 0.1: i.e. least 80% of the CC-I annotators agreed that the text referenced the specified attribute/identity, and at most 10% of them viewed the comment as toxic.

We chose to focus on only non-toxic examples (as rated by the CC-I annotators) in our experiments, because toxic examples can introduce an unwanted confounding factor: there are many examples in the dataset that are only toxic because they contain a slur, and removing or substituting the slur often renders the resulting text non-toxic. Since we are focused on the ability to generate counterfactuals with the same label as the original, we excluded these examples from our dataset. Note that this a choice we make in the context of this particular application, but the general methodology described here could also be used to investigate toxic original examples if deemed appropriate.

4.2 Counterfactual generation

We compare our LLM-D-based generation method to three other methods: ablation, substitution, and the Polyjuice counterfactual generator (Wu et al., 2021). We summarise each of these methods here, and full details are given in Appendix B.

We generate a list of keywords relevant to each topic using frequency analysis on the entire CC-I corpus, followed by manual curation to remove words that often co-occur with a sensitive attribute but are not specific to that topic (e.g. "discriminating" and "surgery" for transgender identity).

To generate ablation counterfactuals, we replace any occurrence of the keywords in our input examples with the empty string. For substitution, we replace all religion-based keywords with a corresponding concept from Christianity, and all sexuality/gender words with their "opposite", e.g. "gay" → "straight", "transgender" → "cisgender". Keywords with no obvious replacement (e.g. "transition", or "Israel") are left unchanged. Note that this can make the substitution method appear artificially good at performing multiple consistent substitutions within a sentence, something that can usually only be achieved with complex rule-based systems (e.g. Lohia (2022)), and comes at the cost of limited counterfactual diversity. This is discussed further in Section 5 when comparing the results of substitution and LLM-D counterfactuals.

In order to fairly compare our LLM-D method with Polyjuice, we generated 16 Polyjuice counterfactuals per input: 8 with no constraints on generation, and 8 where we first used our ablation keyword list to replace all topic-specific keywords in the input sentence with the token `[BLANK]`. These 16 results were then filtered and ranked in the same way as with LLM-D, and the top result returned.

Examples for each method are given in Table 2.

4.3 Counterfactual evaluation

All human annotation of our generated counterfactuals were performed by three of the authors. Each annotator initially rated a subset of the examples, divided to ensure that every counterfactual received at least two ratings, and any examples with non-unanimous scores were passed to the third rater (with scores hidden) for a tiebreaker vote. Examples that received three distinct ratings for a category (yes/unsure/no) were discarded; the only

Method	# examples	Fluent	Attribute ref	Label	FAL and... Meaning 4	Meaning 3+	**Meaning 2+**
Ablation	200	46.6	87.6	99.5	33.0	33.0	33.0
Substitution	200	96.5	88.7	100.0	0.0	84.0	84.5
Polyjuice	162	71.2	15.4	88.8	2.5	4.9	10.5
LLM-D	191	95.7	71.1	96.3	14.1	39.3	62.3

Table 3: Percentage of counterfactuals (generated from Islam-referencing texts) that were labeled by annotators as being fluent, not referencing the sensitive attribute, and having the same label as the original, respectively. "FAL and Meaning n+" lists the percentage of examples that satisfied all of these criteria *and* were given a score of n or higher by annotators for similarity of meaning.

exception to this was the Similarity of Meaning category, where we averaged the raters' scores.

In order to ensure rating consistency and refine the clarity of the instructions, we performed two smaller rounds of test annotation first (50-100 examples) followed by a review session to discuss examples with divergent scores or "unsure" ratings. While the annotators were of diverse genders (male, female, non-binary) and moderately to extremely familiar with the sensitive attributes chosen for our experiments, we also note that they were all white citizens of Western countries and that this could have informed their interpretation of the toxicity task and what substitutions are "neutral".

4.4 Toxicity detection

We use our generated counterfactuals to evaluate the robustness of the Perspective API toxicity classifier to counterfactual perturbations.[5] Perspective API defines toxicity as "a rude, disrespectful, or unreasonable comment that is likely to make you leave a discussion"; the toxicity score is the predicted probability of a reader perceiving the input as toxic.

We focus on the change in predicted toxicity score from original to counterfactual. This is both because any toxicity cut-off threshold will likely vary by use-case, and because we expect that large changes in score will provide interesting and useful information about the classifier even if they do not happen to straddle the toxicity threshold.

5 Results

5.1 Comparison of generation methods

We sample 200 examples that reference Islam from our curated subset of CivilComments-Identities and generate a counterfactual with each of four methods: ablation, substitution, Polyjuice, and our LLM-D-based method. The resulting 753 counterfactuals were shuffled and split between the three annotators for rating;[6] annotators had access to the sensitive attribute label but not the generation method for each example. The results are given in Table 3. Recall that for our purposes, a counterfactual is "good" if it is fluent, does not reference the sensitive attribute, has the same label as the original, and scores at least 2 on similarity of meaning (bolded column in Table 3).

In Table 3 we see that ablation counterfactuals are often not fluent, but that when the input text can be ablated successfully (e.g. sentences where the keywords are used as adjectives, such as "The Muslim woman...") the resulting counterfactuals all receive the maximum score for Similarity of Meaning. Polyjuice was generally unsuccessful at removing references to the sensitive attribute, despite the use of `[BLANK]` tokens to direct the model to the portions of the sentence requiring editing. While substitution achieved higher success rates than LLM-D in this experiment, we show in Section 5.2 below that this may partly have been due to the choice of topic and/or wordlist; this breakdown also does not capture the diversity of topics in the generated counterfactuals.

Finally, we note that the subset of input texts for which ablation produced a good counterfactual tended to be the "easy" examples, in that substitution produced a good counterfactual for 98.5% of this subset, and LLM-D 75%.

5.2 Generation on multiple topics

We sample 100 examples from our curated subset of CivilComments-Identities for each of the attributes Judaism, LGBQ+, and transgender, along with a subset of 100 examples referencing Islam from the set used above. Annotators had access to

[5] www.perspectiveapi.com

[6] Neither LLM-D nor Polyjuice always successfully generated valid counterfactuals, resulting in 191 LLM-D counterfactuals and 162 Polyjuice counterfactuals.

Method	Topic	# examples	Fluent	Attribute ref	Label	FAL and...	
						Meaning 3+	**Meaning 2+**
LLM-D	LGBQ+	99	100.0	54.6	98.6	24.2	48.5
	transgender	99	97.9	43.9	98.5	22.2	36.4
	Judaism	95	96.7	58.4	96.2	41.1	50.5
	Islam	94	95.6	67.0	97.5	35.1	58.5
substitution	LGBQ+	20	93.8	92.3	100.0	50.0	50.0
	transgender	20	95.0	42.1	100.0	35.0	35.0
	Judaism	20	89.5	57.9	100.0	50.0	50.0
	Islam	20	100.0	80.0	100.0	80.0	80.0

Table 4: Percentage of examples satisfying each rating criteria, split by topic. Columns are similar to Table 3.

the sensitive attribute label for each example while rating. Results are given in Table 4.

The key observation here is that our LLM-D-based method generalises easily to multiple topics. We also see further evidence (e.g. attribute reference in Table 4) that our pipeline's automated ranking requires further finetuning, in particular for identifying counterfactuals which have successfully removed all references to the sensitive attribute. In examining discarded LLM-D responses we found that of the 200 examples where the top-ranked LLM-D response did not pass human rating, 105 of these examples (52.5%) had a plausible counterfactual further down the ranking (as judged by one annotator); including these in our evaluation would have raised LLM-D's overall success rate to 75%.

We also generate substitution counterfactuals for a subset of 20 randomly selected examples for each topic, and find that substitution performs almost identically to LLM-D at generating good counterfactuals for each of the non-Islam topics. For the LGBQ+ and transgender categories in particular, this may be due to the fact that explicit labels are most commonly used only to refer to minority groups: one talks about "same-sex marriage" and "transgender athletes", but simply "marriage" and "athletes" when referring to the majority group. Thus a reference to e.g. "cisgender atheletes" still carries an implicit reference to transgender issues. This highlights the need for more complex and diverse counterfactual generation techniques that do not rely solely on substitutions and wordlists.

5.3 Toxicity detection

Throughout this section we restrict our attention only to the "good" counterfactuals (as rated by the human annotators) because poor-quality ones can produce artificially high or low swings in toxicity (due to changing the text too much relative to the original, or by failing to remove the sensitive attribute, respectively); we omit Polyjuice because it produced too few good examples to analyse.

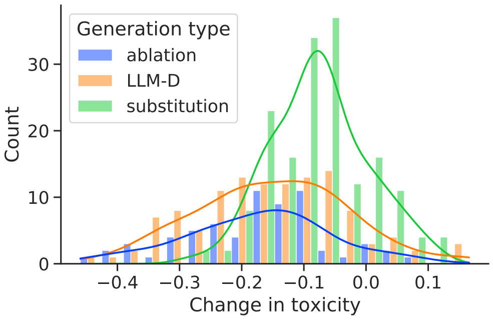


Figure 2: Differences in toxicity score from original texts to their counterfactuals; negative scores indicate that Perspective API rated the counterfactual *less likely* to be viewed as toxic than the original.

Counterfactuals generated by all methods have lower predicted toxicity scores on average than the original Islam-referencing texts, as shown in Figure 2; see also Figure 3 in the appendix for a more detailed breakdown. Substitution produce the smallest change in toxicity scores: an average difference of -0.08, compared to -0.15 for LLM-D and -0.17 for ablation. This suggests that counterfactuals generated by LLM-D and other methods may be producing more challenging examples for the classifier than substitution is, possibly because substitution (by design!) produces text that stays within the same broad topic, and this lack of diversity can make it harder to uncover unexpected negative associations in the classifier.

We also look at the average change in toxicity score across the four topics for both LLM-D and substitution-generated counterfactuals (Table 5). While the sample sizes are too small to draw concrete conclusions, the small average change in toxicity for religion-referencing substitution counter-

Method	Topic	# ex	Avg tox diff
LLM-D	LGBQ+	48	-0.25
	transgender	36	-0.10
	Judaism	48	-0.11
	Islam	55	-0.15
substitution	LGBQ+	10	-0.28
	transgender	7	-0.15
	Judaism	10	-0.04
	Islam	16	- 0.05

Table 5: Average difference in toxicity from original to counterfactual, measured on the good counterfactual pairs generated in Section 5.2. A negative value indicates that the Perspective API classifier found the counterfactual *less* toxic than the original.

factuals compared both to other topics and to LLM-D-generated counterfactuals reinforces the conjecture that the toxicity classifier may view all references to religion as similarly toxic. This suggests that more diverse counterfactuals are indeed necessary to effectively probe a model for subtle counterfactual fairness issues.

Note that the average change in toxicity score is not necessarily meaningful to an end-user. For example, if Perspective API is used to remove comments online with scores above a certain threshold, only score changes around that threshold will have a noticeable end-user impact. Figures 3 and 4 in the appendix provide a more detailed breakdown of how these score changes were distributed, which can help to place the above results in context. However, for the purposes of counterfactual fairness probing we believe it is still important to look at all score changes, not only those near the cut-off point, as this can help to identify areas of potential bias *before* end-users are affected.

6 Conclusion

Our Contributions We have defined a new counterfactual generation task for fairness probing of text classifier models, and have shown that several common types of methods fail to satisfy the requirements of this task and that these failures may limit the effectiveness of the resulting counterfactuals in probing these classifier models. We further show that our LLM-D-based approach combined with automated and human rating can generate high quality, diverse, and complex counterfactual pairs from real-world text examples.

Usage and Limitations Counterfactuals generated via our LLM-D-based approaches could used both to test for undesired behaviour in classifiers and potentially to mitigate that behaviour via methods such as dataset augmentation, as has been found useful in various settings, e.g. Dinan et al. (2020), Hall Maudslay et al. (2019).

However, we emphasise that this is not without risk. Language models are known to produce toxic text (Wallace et al., 2019) and reflect or amplify biases from their training data (Sheng et al., 2019b), among other problems (Bommasani et al., 2021, §5); we always recommend human review on at least a subset of the data when using potentially sensitive generated text. Language is contextual, and there is a great deal of social context that must be accounted for when attempting to evaluate the behaviours and biases of machine learning models and generated text, so it is important for human review to be performed by a diverse pool of reviewers knowledgeable about the downstream task and the social issues at play (in contrast to the small set of annotators for this illustrative study).

Using text generated from methods such as ours is also not appropriate in all situations. For example, we emphasise that this generated data should be used to *augment* other forms of data, not replace it. Similarly, while this study sought to generate diverse texts for analysis, a more restrictive definition of counterfactual may be appropriate when using generated text to *mitigate* classifier issues, e.g. by using a stricter cut-off for the "Similarity of Meaning" evaluation criteria.

Future Research There are several areas of future research to highlight. Most generally, for this investigation we focused on one way this framework can be useful, and made several narrowing choices; however, our framework can be useful in other contexts and applications such as investigating false negatives (by considering original examples that are toxic), probing other types of classifiers than toxicity models, or generating other types of counterfactuals than simply removing the sensitive attribute (e.g. rewording text to explore model robustness). Furthermore, our method would benefit from improved control over the LLM-generated text through e.g. prompt tuning (Lester et al., 2021), demonstration-based prompt-engineering and adversarial decoding (Hartvigsen et al., 2022), or fine-tuning (Wei et al., 2021), as well as more effective filtering of counterfactuals that still reference the sensitive attribute.

Acknowledgements

We would like to thank Vinodkumar Prabhakaran, Ian Tenney, Emily Reif, Ian Kivlichan, Lucy Vasserman, Blake Lemoine, Alyssa Chvasta, and Kathy Meier-Hellstern for helpful discussions and feedback at every stage of this project.

References

Abubakar Abid, Maheen Farooqi, and James Zou. 2021. *Persistent Anti-Muslim Bias in Large Language Models*, page 298–306. Association for Computing Machinery, New York, NY, USA.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the opportunities and risks of foundation models.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced metrics for measuring unintended bias with real data for text classification. *CoRR*, abs/1903.04561.

Luke Breitfeller, Emily Ahn, David Jurgens, and Yulia Tsvetkov. 2019. Finding microaggressions in the wild: A case for locating elusive phenomena in social media posts. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1664–1674, Hong Kong, China. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level BLEU. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367, Baltimore, Maryland, USA. Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Aida Mostafazadeh Davani, Ali Omrani, Brendan Kennedy, Mohammad Atari, Xiang Ren, and Morteza Dehghani. 2021. Improving counterfactual generation for fair hate speech detection. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 92–101, Online. Association for Computational Linguistics.

Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial bias in hate speech and abusive language detection datasets. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 25–35, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Angela Fan, Adina Williams, Jack Urbanek, Douwe Kiela, and Jason Weston. 2020. Queens are powerful too: Mitigating gender bias in dialogue generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8173–8188, Online. Association for Computational Linguistics.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, page 67–73, New York, NY, USA. Association for Computing Machinery.

Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed H Chi, and Alex Beutel. 2019. Counterfactual fairness in text classification through robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 219–226.

Rowan Hall Maudslay, Hila Gonen, Ryan Cotterell, and Simone Teufel. 2019. It's all in the name: Mitigating gender bias with name-based counterfactual data substitution. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5267–5275, Hong Kong, China. Association for Computational Linguistics.

Xiaochuang Han and Yulia Tsvetkov. 2020. Fortifying toxic speech detectors against veiled toxicity. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7732–7739, Online. Association for Computational Linguistics.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. Toxigen: Controlling language models to generate implied and adversarial toxicity. In *ACL*.

Ben Hutchinson, Vinodkumar Prabhakaran, Emily Denton, Kellie Webster, Yu Zhong, and Stephen Denuyl. 2020. Social biases in NLP models as barriers for persons with disabilities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5491–5501, Online. Association for Computational Linguistics.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Hannah Rose Kirk, Bertram Vidgen, Paul Röttger, Tristan Thrush, and Scott A. Hale. 2021. Hatemoji: A test suite and adversarially-generated dataset for benchmarking and detecting emoji-based hate. *CoRR*, abs/2108.05921.

Brian Larson. 2017. Gender as a variable in natural-language processing: Ethical considerations. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 1–11, Valencia, Spain. Association for Computational Linguistics.

Alyssa Lees, Daniel Borkan, Ian Kivlichan, Jorge Nario, and Tesh Goyal. 2021. Capturing covertly toxic speech via crowdsourcing. In *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, pages 14–20, Online. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pranay Lohia. 2022. Counterfactual multi-token fairness in text classification. *arXiv preprint arXiv:2202.03792*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. 2020. Generate your counterfactuals: Towards controlled counterfactual generation for text. *arXiv preprint arXiv:2012.04698*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Vinodkumar Prabhakaran, Ben Hutchinson, and Margaret Mitchell. 2019. Perturbation sensitivity analysis to detect unintended model biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5740–5745, Hong Kong, China. Association for Computational Linguistics.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2021. A recipe for arbitrary text style transfer with large language models. *arXiv e-prints*, pages arXiv–2109.

Alexis Ross, Ana Marasović, and Matthew E Peters. 2021a. Explaining nlp models via minimal contrastive editing (mice). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3840–3852.

Alexis Ross, Tongshuang Wu, Hao Peng, Matthew E Peters, and Matt Gardner. 2021b. Tailor: Generating and perturbing text with semantic controls. *arXiv preprint arXiv:2107.07150*.

Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet Pierrehumbert. 2021. HateCheck: Functional tests for hate speech detection models. In *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)*, pages 41–58, Online. Association for Computational Linguistics.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Proceedings of NAACL-HLT*, pages 8–14.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 2020. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5477–5490, Online. Association for Computational Linguistics.

Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. 2019a. The woman worked as a babysitter: On biases in language generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3407–3412.

Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. 2020. Towards controllable biases in language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3239–3254.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019b. The woman worked as a babysitter: On biases in language generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3407–3412, Hong Kong, China. Association for Computational Linguistics.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Tianlu Wang, Xuezhi Wang, Yao Qin, Ben Packer, Kang Li, Jilin Chen, Alex Beutel, and Ed Chi. 2020. Cat-gen: Improving robustness in nlp models via controlled adversarial text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5141–5146.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *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)*, pages 6707–6723.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Xuhui Zhou, Maarten Sap, Swabha Swayamdipta, Yejin Choi, and Noah Smith. 2021. Challenges in automated debiasing for toxic language detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3143–3155, Online. Association for Computational Linguistics.

A LLM-D counterfactual text generation

A.1 Setup

Following Reif et al. (2021), we use "{" and "}" delimiters in the formatting of the prompt to encourage LLM-D to provide its response in a similar format, and automatically discard any text outside of the first set of delimiters in each response. The prompts are formatted in a second-person conversational style, as this is the style of data that LLM-D was finetuned on; for a template suitable for standard next-token language models, see (Reif et al., 2021, Table 7).

While our initial experiments used the underlying General Language Model (GLM) part of LLM-D, all results in this paper were generated using

LLM-D. We used a temperature of 1 and $k = 40$ for the top-k next token sampling, and we did not discard responses regardless of the "safety" score LLM-D assigned to them, as we found that this too severely curtailed the diversity of responses. We mitigated the safety risks of this by ensuring that we had a robust human evaluation step in place later in the pipeline.

We also filter out responses with failure modes observed often in early experiments, including responses that were just a string of punctuation or the "shrug emoji" ¯_(ツ)_/¯, verbatim repetitions of the input text, and responses that regurgitated part of the prompt ("here is a rewrite...", "here is some text..."). These filters were applied to the initial set of 16 responses from the model.

A.2 Prompt and instruction selection

The full set of prompts used in all of our experiments are listed in Table 8; these are the same prompts used in Reif et al. (2021).

We experimented with different prompts, but found that more task-specific prompts did not produce measurably better results, and in fact found that LLM-D tended to overfit much more strongly to the final few prompts when the prompts specifically referenced the sensitive attribute. For example, using a set of 7 prompts demonstrating examples of counterfactual generation specifically for transgender identity, where the last two prompts referenced beauty pageants and Kiwi transgender weightlifter Laurel Hubbard respectively, the (unfiltered) LLM-D responses to the 100 transgender-referencing examples used in Section 5.2 included 22 references to New Zealand, 31 references to weight lifters, and 5 references to beauty queens / beauty pageants. By comparison, using the prompts in Table 8 generated 0 results involving any of these keywords, and a total of 7 results referencing bells/snow/trees (see the final two prompts in Table 8).

For the rewriting instruction, we found that "make this not about [sensitive attribute]" helped to focus the language model's attention on the desired parts of the sentence (as opposed to Polyjuice, which would often produce permutations that were completely unrelated to the sensitive attribute reference in the sentence) but that this did not reliably translate into *removing* the reference to the sensitive attribute. However, the fact that LLM-D produces 16 independent responses meant that there was consistently at least one response that did satisfy the criteria to be a good counterfactual, and one direction of future work is to automatically identify these responses more effectively.

A.3 Automated metrics

We used the implementation of BLEU score provided by the `sacrebleu` package, using the NIST smoothing method as described in Chen and Cherry (2014) to mitigate the fact that we are using a corpus-level metric to compute scores on individual pairs of sentences.

The implementation of BERTScore is the one provided by the authors (Zhang et al., 2020), modified to accept a Flax-based BERT model. BERTScore computes both a recall score (which rewards text pairs where everything in the original sentence is also represented in the counterfactual) and a precision score (rewards pairs where everything in the counterfactual is also represented in the original).[7] We use the resulting F1 score as our metric since we want our counterfactuals to neither add too much nor delete too much compared to the original text.

The attribute classifier is also JAX/Flax-based, and comprises a 2-layer fully connected network (hidden dimension 2048), using the first token of the input text's BERT representation (Devlin et al., 2019) as the embedding function. It was trained on a subset of CivilComments-Identities (all texts, regardless of toxicity, that referenced at least one attribute of interest with a score > 0.5, along with 20k negative examples that referenced none of the attributes of interest) using the AdamW optimizer (Loshchilov and Hutter, 2018) (with learning rate 0.001, weight decay 0.002) for 36k steps with a batch size 256, using a binary cross-entropy loss function to allow for multi-label predictions.

B Counterfactual generation methods

B.1 Ablation

For ablation, we generate a list of key terms per identity and simply remove those terms from each text. The lists for each attribute are in Table 6.

The term list was generated by fitting a unigram naive bayes classifier to the non-toxic subset of Civil Comments data (toxicity < 0.1), separating texts labeled with the given identity group (attribute score > 0.5) from a random sample of the rest. The

[7] Note that these are not symmetric: for example, a counterfactual that simply repeats the original text but adds an extra detail to the end would score more highly on recall than precision.

20 features (unigrams) most strongly associated with the identity class were used as the candidate wordlist, and were filtered by hand to remove irrelevant terms.

We emphasise that these wordlists are *not* complete representations of the corresponding attributes and that our ablation counterfactuals were generated purely to provide a baseline score for comparison to other methods.

B.2 Substitution

To generate counterfactuals using substitution, we take the ablation wordlists and (where possible) assign each one a corresponding word from another identity in the same broad category, e.g. replacing one religion with another. For examples with no plausible substitution (e.g. "transition" in the transgender category) we leave the word unchanged. See Table 7 for the full set of word pairs.

As with the ablation wordlists above, we emphasise that these are not necessarily complete representations of the corresponding attributes. They were generated purely for the purposes of providing a baseline for comparison in our experiments, and should not be used as-is to generate counterfactuals for fairness probing in real world settings.

B.3 LLM-D

We use the same fixed set of prompts for every input text (see Appendix A.2 and Table 8), and the instruction "make this not about X", where X is the sensitive attribute referenced in the input text. LLM-D generates up to 16 responses per input, which are filtered as described in Appendix A.3 and then ranked by taking the average of their BLEU score and BERTScore F1 score. Only the top-ranked example is returned for rating.

For some inputs, it can happen that none of LLM-D's responses are of sufficient quality to pass the filtering step. We rerun the generation pipeline on each of these failed inputs until a counterfactual is returned, up to a maximum of 5 attempts.

B.4 Polyjuice

Polyjuice (Wu et al., 2021) is a general-purpose counterfactual generator that uses a finetuned LM (GPT-2) along with control-codes to generate diverse permutations of sentences. To our knowledge, Polyjuice has not been evaluated for fairness probing, but its flexible generation abilities make it a promising approach to compare with.

A Polyjuice user can choose from various types of permutation (negation, shuffle, deletion, etc) and can even specify where in the sentence the edit should be made by replacing words or phrases with the `[BLANK]` token.

For each input text, we generate 16 potential counterfactuals: 8 where we allow Polyjuice to choose which parts of the text to modify, and 8 where we direct its attention to the sensitive attribute reference(s) by replacing all keywords from the corresponding ablation list with the `[BLANK]` token. These 16 examples are then filtered and ranked by the same criteria as the LLM-D examples, and the top-scoring one is returned for rating.

As with the LLM-D counterfactuals, we rerun the Polyjuice generation pipeline on inputs that failed to return a valid counterfactual, up to 5 times per example.

We found that some of the generated Polyjuice counterfactuals still contained tokens such as `BLANK` or `EMPTY`, which we replaced with the empty string before rating the examples.

C Human rater guidelines

We rate each (original, counterfactual) pair on four criteria:

- fluency
- presence of sensitive attribute
- similarity of meaning
- similarity of label

Precise definitions of each criteria (along with examples) are given in the rater instructions below. For our purposes, we required an (original, counterfactual) pair to be rated "yes" for fluency, "no reference" for presence of sensitive attribute, "yes" for similarity of label, and at least 2 (out of 4) for similarity of meaning.

These guidelines are designed to be flexible and easily modifiable to apply to multiple text formats (internet comments, video titles, etc) and downstream tasks (toxicity detection, sentiment analysis, etc). We highlight the parts that are specific to our task by [bracketing and underlining] them.

C.1 Rater instructions

1. ***Fluency/consistency***
   Is the text fluent and consistent, and does it make sense? (yes/no/unsure)
   Ignore any extraneous punctuation marks at the beginning or end of the text. You should also ignore capitalisation / lack thereof unless

it makes a significant difference to the fluency. Evaluate the fluency relative to the original text and in the context of the domain: these examples are [internet comments], so they need not adhere to perfect spelling and grammar to be realistic examples.

Examples of sufficiently fluent sentences:

- “islam is great, its just a fact” [fine: minor incorrect grammar but not unrealistic for internet speech]
- “Some folks are born white... just accept it.” [fine: awkward phrasing but not inconsistent]

Examples of sentences that are not consistent or do not make sense:

- “*Christians* want to impose *sharia law* on the country.” [inconsistent: Sharia law is associated with Islam]
- “The problem with on violence is it harms” [does not make sense]
- “—!!!!!–,,,,,,xxxxz” [gibberish]

2. ***Presence of sensitive attribute***

   Does the text:

   - **explicitly reference the attribute?**
   - **implicitly reference the attribute?**
   - **not reference the attribute?**

   The targeted attribute will be specified for each example; references to other sensitive attributes (e.g. references to other religions when the specified attribute is Islam) do not count.

   An *implicit reference* is a group or identity referenced but not stated explicitly. For example:

   - “Jack went on a date with his boyfriend” implicitly references the fact that Jack is queer but does not state it explicitly.
   - “Our state is trying to pass a bathroom bill, this sucks” implicitly references transgender identity, since “bathroom bill” is commonly understood to be a type of law that targets trans and non-binary people.
   - “She wore a hijab on stage” is an implicit reference to Islam, since the hijab is a common marker of Muslim identity.

   If the text contains an explicit reference to the attribute, there’s no need to evaluate it for implicit references as well.

3. ***Similarity of meaning***

   Taking into account both the changes made to remove references to the sensitive attribute and any other changes in the sentence, how similar is the text to the original? (scale of 0 to 4)

   - **Score of 4**: Retains the same grammatical structure and overall meaning as the original, and changes only the parts of the sentence required to remove the sensitive attribute (no other changes or deletions).
   - **Score of 3**: Substitutes or removes the sensitive attribute, only differing from the original in ways that don’t materially affect the meaning of the example, e.g. minor grammatical changes, use of synonyms, substitutions to similar or semantically neutral concepts (“BLM” → “environmental protestors”, “Allah” → “God” are examples of similar substitution, “Muslims” → “people”, “Antifa” → “protest movement” are examples of neutral substitution).

     Note that here “neutral” means words like “people” ,“citizens”, “protest movement”, “religion” etc. Non-toxic substitutions that change the meaning of the sentence should be assigned a score of 2 (see below).
   - **Score of 2**: Makes an additional change (addition or removal) beyond the minimal changes required to remove the sensitive attribute, and/or makes edits that change the meaning of the sentence.
   - **Score of 1**: Bears some resemblance to the original sentence, but has major changes beyond those necessary to remove the sensitive attribute.
   - **Score of 0**: Is completely unrelated to the original sentence.

Here are some example texts for each score.

- Original:
  - How would you feel about opening a meeting with a Muslim prayer to Allah?
- Score of 4:
  - How would you feel about opening a meeting with a prayer? [no changes beyond minimal necessary]
  - How would you feel about opening a meeting with prayers?
- Score of 3:
  - How would you feel about opening a

    meeting with a *Christian* prayer to *God*? [substitution to similar concept]
  - How would you feel about opening a meeting with a *religious reading*? [substitution to neutral concept]
  - How would you feel about opening *this event* with a prayer? [minor extra rewording beyond minimum necessary]
- Score of 2:
  - *Since we're in Portland*, how would you feel about opening this meeting with a prayer? [addition of extra detail in addition to necessary chances]
  - How would you feel about opening a meeting with a *brief political statement*? [substitution that changes the meaning of the sentence]
  - How would you feel about opening a meeting? [removal of extra detail that changes the meaning in a non-trivial way]
- Score of 1:
  - What's all this about meetings and prayers? [similar topic but major changes to original beyond those necessary to remove attribute]
- Score of 0:
  - By the banks of the Seine. [no resemblance to original]

4. ***Similarity of label***
   Should this example be assigned the same ground-truth [toxicity] label as the original text? (yes/no/unsure)
   Note: you are not trying to determine whether the two texts are equivalent or whether the concepts referenced in either text are comparable or even true: *your answer should depend only on whether you would think it would cause harm if a [toxicity] classifier model treated the two texts differently.*
   [For the purposes of this question, we define toxicity to mean "a rude, disrespectful, or unreasonable comment that is likely to make you leave a discussion".][8]
   The following are examples where both texts *should* be assigned the same label, because a [toxicity] classifier model should treat them the same way:
   - Text: Miss Nevada (2021) is a beauty queen.
     Original: Miss Nevada (2021) is a transgender woman.
     (both non-toxic)
   - Text: Maybe the lead scientist is a woman.
     Original: Maybe the lead scientist is gay.
     (both non-toxic)
   - Text: I hate white people.
     Original: I hate straight people.
     (both toxic)

   The following are examples where the example should *not* be assigned the same label as the original text, because a [toxicity] classifier model should *not* treat them the same way:
   - Text: All white men are dangerous. (toxic)
     Original: All white men think cats are great. (non-toxic)
   - Text: Can't trust a velociraptor, they don't have morals. (non-toxic)
     Original: Can't trust a Christian, they don't have morals. (toxic)
5. ***Reject for other reason? (yes/leave blank)***
   Use this field to flag examples that pass all of the other criteria but seem inappropriate for use as counterfactuals.

C.2 Example

Text: He was a baker? I did not see that in the story.
Original: He was Muslim? I did not see that in the story.
Sensitive attribute: Islam

1. Is the text fluent and consistent, and does it make sense? (yes/no/unsure)
   `Yes`
2. Does the text: explicitly reference the attribute? implicitly reference the attribute? not reference the attribute?
   `Not reference the attribute`
3. Taking into account both the changes made to remove references to the sensitive attribute and any other changes in the sentence, how similar is the text to the original? (scale of 0 to 4)
   `2`

[8] This is the definition of toxicity used by Perspective API; https://developers.perspectiveapi.com/s/about-the-api-key-concepts

4. Should this example be assigned the same ground-truth toxicity label as the original text? (yes/no/unsure)
   `Yes`
5. Reject for other reason? (yes or ⟨leave blank⟩)
   ⟨`leave blank`⟩

Attribute	Ablation unigrams
LGBQ+	gay, gays, homosexual, homosexuality, homosexuals, lesbian, lesbians, lgbt, lgbtq, sexuality
transgender	bathroom, bathrooms, hormones, lgbt, lgbtq, testosterone, trans, transgender, transgendered, transition
Judaism	holocaust, israel, israeli, israelis, jew, jewish, jews, judaism, semitic, semitism, zionist
Islam	allah, hijab, islam, islamic, islamist, islamists, islamophobia, koran, mosque, mosques, muslim, muslims, quran, sharia, sunni

Table 6: Ablation Wordlists, generated as described in Appendix B.1. Note that these are not intended to be comprehensive wordlists for each topic, nor are all of the words direct references to the attribute itself (e.g. "Israel" or "bathroom"); we chose to retain these indirect references if they appeared in the top 20 unigrams produced by the naive Bayes classifier since we were evaluating the resulting counterfactuals on implicit references to the attribute as well as explicit ones.

Attribute	Replacement category	Substitution wordpairs
LGBQ+	heterosexual	(gay, straight), (gays, straights), (homosexual, heterosexual), (homosexuality, heterosexuality), (homosexuals, heterosexuals), (lesbian, straight), (lesbians, straights), (lgbt, straight), (lgbtq, straight)
transgender	cisgender	(lgbt, cis), (lgbtq, cis), (trans, cis), (transgender, cisgender), (transgendered, cisgendered)
Judaism	Christianity	(jew, christian), (jewish, christian), (jews, christians), (judaism, christianity)
Islam	Christianity	(allah, god), (hijab, cross), (islam, christianity), (islamic, christian), (islamist, fundamentalist), (islamists, fundamentalists), (islamophobia, anti-christian bias), (koran, bible), (mosque, church), (mosques, churches), (muslim, christian), (muslims, christians), (quran, bible), (sharia, canon law), (sunni, catholic)

Table 7: Substitution Wordlists. Note that while some of the pairings are direct analogs (e.g. "gay" → "straight", "Muslim" → "Christian"), others were chosen to maximise the chances of generating valid counterfactuals while retaining the general meaning of the sentence (e.g. "LGBTQ" → "straight"/"cis", "hijab" → "cross"); we are *not* implying that all of these pairings are completely analogous.

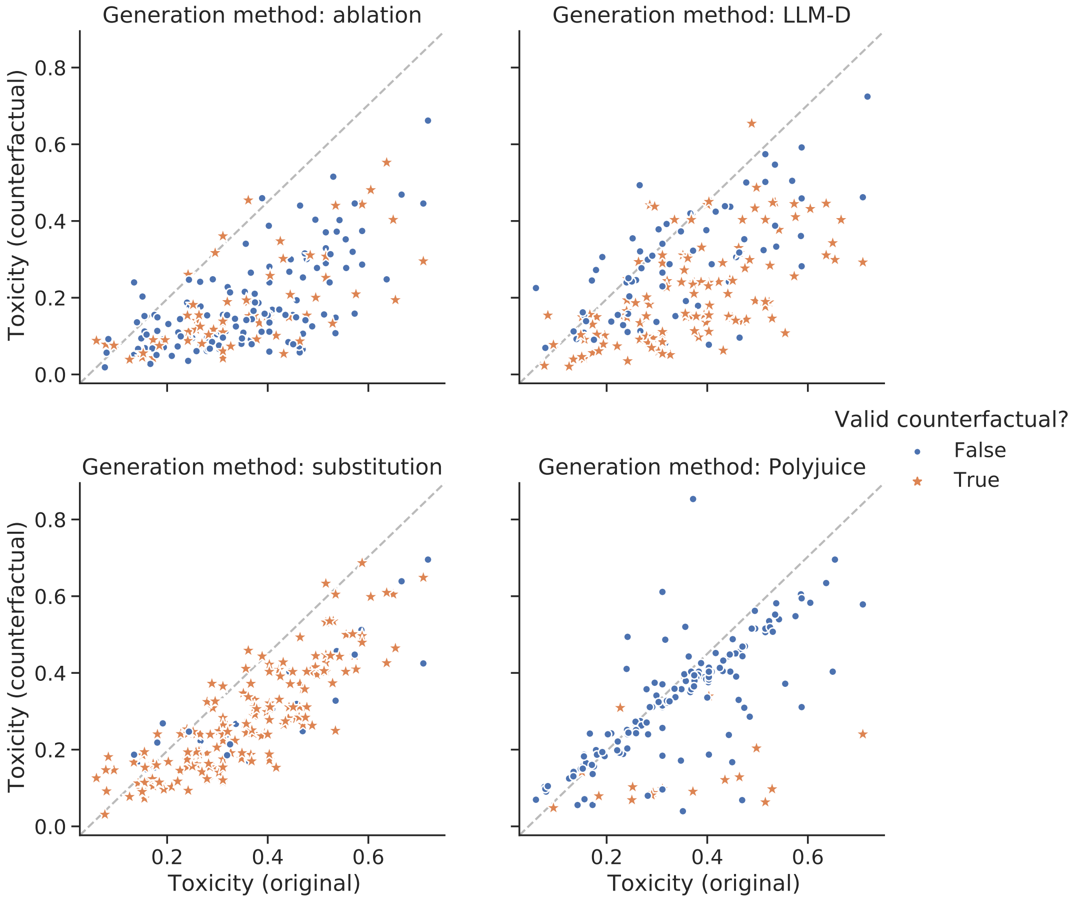


Figure 3: Toxicity scores of counterfactuals (generated from the Islam-referencing texts in Section 5.1) plotted against the toxicity scores of their original text; points in the lower right portion of each graph correspond to examples where Perspective API rated the counterfactual as *less* likely to be toxic than the original. We include the counterfactuals that did not pass the human rating step in order to illustrate the effects of different counterfactual generation methods on toxicity detection: for example, ablation failed mostly on the fluency criteria so its “poor” counterfactuals still exhibit a drop in toxicity here, whereas Polyjuice failed mostly on removing references to the sensitive attribute so its “poor” counterfactuals tend to cluster around the $y = x$ line.

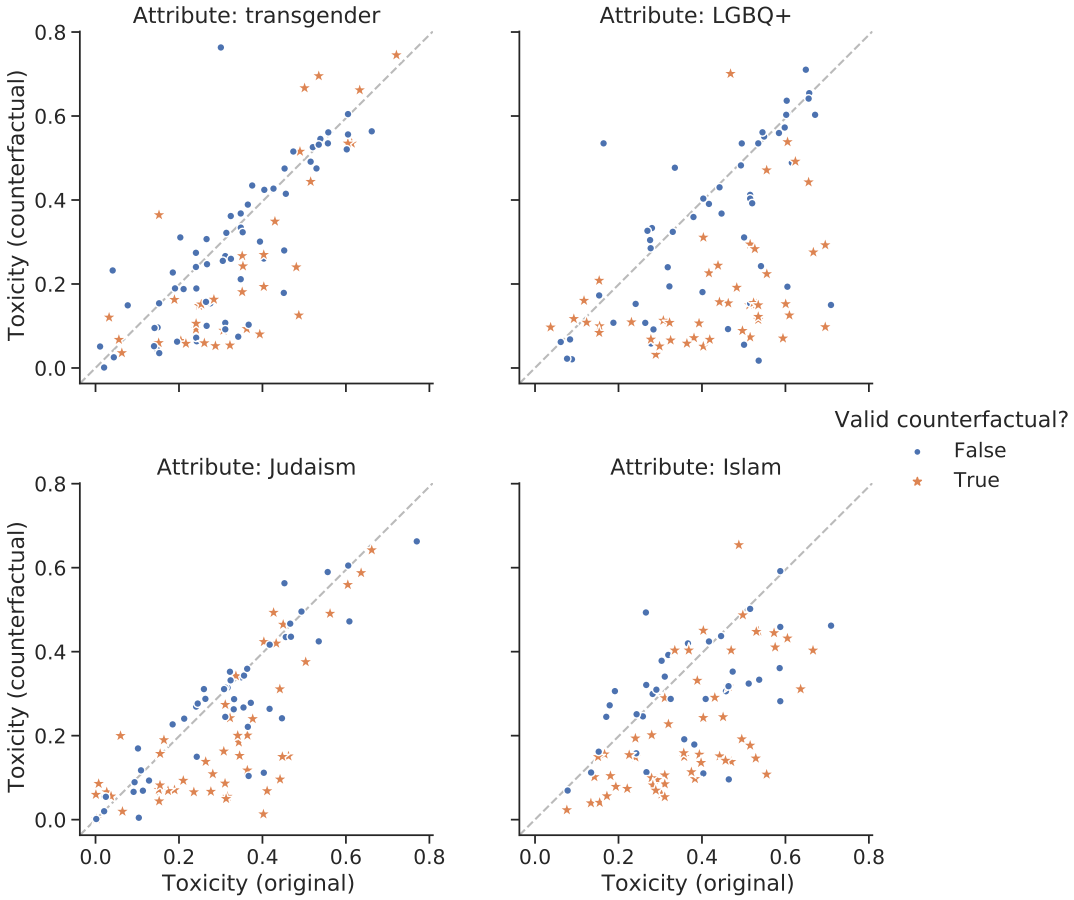


Figure 4: Toxicity scores of LLM-D counterfactuals (from Section 5.2) plotted against the toxicity scores of their original text; points in the lower right portion of the graph correspond to examples where Perspective API rated the counterfactual as *less* likely to be toxic than the original. Observe that the LGBQ+ category sees some swings in toxicity score that could result in a label flip (represented by the points closest to the bottom right corner), a phenomenon which is much less common with the other three topics.

0: Here is some text: {When the doctor asked Linda to take the medicine, he smiled and gave her a lollipop.}. Rewrite it to be more scary.

1: {When the doctor told Linda to take the medicine, there had been a malicious gleam in her eye that Linda didn't like at all.}

0: Here is some text: {they asked loudly, over the sound of the train.}. Rewrite it to be more intense.

1: {they yelled aggressively, over the clanging of the train.}

0: Here is some text: {When Mohammed left the theatre, it was already dark out}. Rewrite it to be more about the movie itself.

1: {The movie was longer than Mohammed had expected, and despite the excellent ratings he was a bit disappointed when he left the theatre.}

0: Here is some text: {next to the path}. Rewrite it to be about France.

1: {next to la Siene}

0: Here is some text: {The man stood outside the grocery store, ringing the bell.}. Rewrite it to be about clowns.

1: {The man stood outside the circus, holding a bunch of balloons.}

0: Here is some text: {the bell ringing}. Rewrite it to be more flowery.

1: {the peales of the jangling bell}

0: Here is some text: {against the tree}. Rewrite it to be include the word "snow".

1: {against the snow-covered bark of the tree}'

0: Here is some text: {**input text here**}. Rewrite it to be **instruction here**.

Table 8: The full prompt text used to generate counterfactuals with LLM-D. The "0" and "1" indicate two speakers, since LLM-D expects inputs formatted in turns of conversation. The text to be rewritten and the corresponding instruction (e.g. "not about transgender people") are inserted in the last line (blue, boldface).