Necessity and Sufficiency for Explaining Text Classifiers: A Case Study in Hate Speech Detection

Esma Balkır, Isar Nejadgholi, Kathleen C. Fraser, and Svetlana Kiritchenko
National Research Council Canada
Ottawa, Canada
{Esma.Balkir,Isar.Nejadgholi,Kathleen.Fraser,Svetlana.Kiritchenko}@nrc-cnrc.gc.ca

Abstract

We present a novel feature attribution method for explaining text classifiers, and analyze it in the context of hate speech detection. Although feature attribution models usually provide a single importance score for each token, we instead provide two complementary and theoretically-grounded scores – *necessity* and *sufficiency* – resulting in more informative explanations. We propose a transparent method that calculates these values by generating explicit perturbations of the input text, allowing the importance scores themselves to be explainable. We employ our method to explain the predictions of different hate speech detection models on the same set of curated examples from a test suite, and show that different values of necessity and sufficiency for identity terms correspond to different kinds of false positive errors, exposing sources of classifier bias against marginalized groups.

1 Introduction

Explainability in AI (XAI) is critical in reaching various objectives during a system's development and deployment, including debugging the system, ensuring its fairness, safety and security, and understanding and appealing its decisions by end-users (Vaughan and Wallach, 2021; Luo et al., 2021).

A popular class of local explanation techniques is feature attribution methods, where the aim is to provide scores for each feature according to how important that feature is for the classifier decision for a given input. From an intuitive perspective, one issue with feature attribution scores is that it is not always clear how to interpret the assigned importance in operational terms. Specifically, saying that a feature is 'important' might translate to two different predictions. The first interpretation is that if an important feature value is changed, then the prediction will change. The second interpretation is that, as long as the feature remains, the prediction will not change. The former interpretation corresponds to the *necessity* of the feature value, while the latter corresponds to its *sufficiency*.

To further illustrate the difference between necessity and sufficiency, we take an example from hate speech detection. Consider the utterance "I hate women". For a perfect model, the token 'women' should have low sufficiency for a positive prediction, since merely the mention of this identity group should not trigger a hateful prediction. However, this token should have fairly high necessity, since a target identity is required for an abusive utterance to count as hate speech (e.g., "I hate oranges" should not be classified as hate speech). In this paper, we develop a method to estimate the necessity and sufficiency of each word in the input, as explanations for a binary text classifier's decisions.

Model-agnostic feature attribution methods like ours often perturb the input to be explained, obtain the predictions of the model for the perturbed instances, and aggregate the results to make conclusions about which input features are more influential on the model decision. When applying these methods to textual data, it is common to either drop the chosen tokens, or replace them with the mask token for those models that have been trained by fine-tuning a masked language model such as BERT (Devlin et al., 2019). However, deleting tokens raises the possibility that a large portion of the perturbed examples are not fluent, and lie well outside the data manifold. Replacing some tokens with the mask token partially remedies this issue, however it raises others. Firstly, the explanation method ceases to be truly model-agnostic. Secondly, a masked sentence is in-distribution for the pre-trained model but out-of-distribution for the fine-tuned model, because the learned manifolds deviate from those formed during pre-training in the fine-tuning step.

To avoid these problems we use a generative model to *replace* tokens with most probable n-

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2672 - 2686

grams. Generating perturbations in this way ensures that the perturbed instances are close to the true data manifold. It also provides an additional layer of transparency to the user, so they can decide whether to trust the explanation by checking how reasonable the perturbed examples seem.

Although supervised discriminative models rely fundamentally on correlations within the dataset, different models might rely on different correlations more or less depending on model architecture and biases, training methods, and other idiosyncrasies. To capture the distinction between correlations in the data and the direct causes of the prediction, we turn to the notion of *interventions* from causal inference (Pearl, 2009). Previous work employing causal definitions of necessity and sufficiency for XAI have assumed tabular data with binary or numerical features. The situation in NLP is much more complex, since each feature is a word in context, and we have no concept of 'flipping' or 'increasing' feature values (as in binary data and numerical data, respectively). Instead, our method generates perturbations of the input text that have high probability of being fluent while minimizing the probability that the generated text will also be a direct cause of the prediction we aim to explain.

As our application domain we choose hate speech detection, a prominent NLP task with significant social outcomes (Fortuna and Nunes, 2018; Kiritchenko et al., 2021). It has been shown that contemporary hate speech classifiers tend to learn spurious correlations, including those between identity terms and the positive (hate) class, which can result in further discrimination of already marginalized groups (Dixon et al., 2018; Park et al., 2018; Garg et al., 2019). We apply our explainability metrics to test classifiers' fairness towards identity-based groups (e.g., women, Muslims). We show how necessity and sufficiency metrics calculated for identity terms over hateful sentences can explain the classifier's behaviour on non-hateful statements, highlighting classifiers' tendencies to over-rely on the presence of identity terms or to ignore the characteristics of the object of abuse (e.g., protected identity groups vs. non-human entities).

The contributions of this work are as follows:

- We present the first methodology for calculating necessity and sufficiency metrics for text data as a feature attribution method. Arguably, this dual explainability measure is more informative and allows for deeper insights into a model's inner workings than traditional single metrics.
- We use a generative model for producing input perturbations to avoid the out-of-distribution prediction issues that emerge with token deletion and masking techniques.
- To evaluate the new methodology, we apply it to the task of explaining hate speech classification, and demonstrate that it can detect and explain biases in hate speech classifiers.

We make the implementation code freely available to researchers to facilitate further advancement of explainability techniques for NLP.[1]

2 Background and Related Work

Explanations are often categorized as to whether they are for an individual prediction (local) or for the model reasoning as a whole (global), and whether the explanation generation is a part of the prediction process (self-explaining) or generated through additional post-processing (post-hoc) (Guidotti et al., 2018; Adadi and Berrada, 2018). The necessity and sufficiency explanations presented here belong to the class of local explanation methods, as do most of the XAI methods applied to NLP data (Danilevsky et al., 2020). It is also a post-hoc method to the degree that it is entirely model-agnostic: all it requires is binary predictions on provided inputs.

There are a few classes of popular techniques for explaining natural language processing models (see Danilevsky et al. (2020) for a survey). One approach is *feature attribution methods* that allocate importance scores to each feature. These can be architecture-specific (Bahdanau et al., 2015; Sundararajan et al., 2017), or model-agnostic (Ribeiro et al., 2016; Lundberg and Lee, 2017).

Another approach is *counterfactual explanations*, which provide similar examples to the input in order to show what kinds of small differences affect the prediction of the model (Wu et al., 2021; Kaushik et al., 2021; Ribeiro et al., 2020; Ross et al., 2020). These contrastive examples are related to the concept of counterfactual reasoning from the causality literature, that formalizes the question: *"Would the outcome have happened if this event had not occurred?"* in order

[1] https://github.com/esmab/necessity-sufficiency

to determine whether the event was a cause of the observed outcome (Pearl, 2009). Counterfactual explanation methods are often targeted at certain semantic or syntactic phenomena such as negation (Kaushik et al., 2021) or swapping objects and subjects (Zhang et al., 2019), and hence do not guarantee that the counterfactuals cover the data distribution around the input text well.

In this work, we combine methods from feature attribution and counterfactual generation models. This allows us to calculate scores that capture local feature importance, and provide counterfactual examples as justification for the assigned scores.

Necessity and sufficiency. These are two notions from causal analysis that capture what one intuitively expects a true cause of an event to exhibit (Pearl, 2009; Halpern, 2016). Several works have recently suggested applying necessity and sufficiency to explain model predictions. Mothilal et al. (2021) used the *actual causality* framework of Halpern (2016) to calculate necessity and sufficiency scores for tabular data. Galhotra et al. (2021) suggested an approach to capture the notions of necessity and sufficiency from probabilistic causal models (Pearl, 2009). Watson et al. (2021) presented a different method for quantifying necessity and sufficiency over subsets of features. We follow the framework of probabilistic causal models, and adopt the definitions from Galhotra et al. (2021). In NLP explanations, necessity and sufficiency have been used for evaluating rationales (Zaidan et al., 2007; DeYoung et al., 2020; Mathew et al., 2021)[2], however to the best of our knowledge, this is the first work to explore their usage for estimating feature attribution scores.

The out-of-distribution problem in feature attribution models. Virtually all model-agnostic feature attribution models calculate importance scores by perturbing input features and assign importance according to which feature changes the outcome the most. However, an issue has been raised that these perturbed inputs are no longer drawn from the data distribution that the model would naturally encounter for a given task (Fong and Vedaldi, 2017; Chang et al., 2018; Hooker et al., 2019; Janzing et al., 2020; Hase et al., 2021). This is problematic because then, any change in the model predictions could be caused by the distribution shift rather than the removal of feature values (Hooker et al., 2019). Recently, Hase et al. (2021) have argued that the problem is due to *social misalignment* (Jacovi and Goldberg, 2021), where the information communicated by the model differs in non-intuitive ways from the information people expect.

One solution to address these issues is to calculate importance scores by marginalizing over counterfactuals that respect the data distribution. Kim et al. (2020) and Harbecke and Alt (2020) adopted this approach and targeted text data specifically by marginalizing over infills generated by BERT. In our preliminary experiments, this resulted in the model putting an overwhelmingly high probability mass to one or few very common words, making the generated perturbations relatively non-diverse.[3] As Pham et al. (2021) also pointed out, BERT is very good at guessing the masked word, doing so correctly about half of the time. This behaviour results in assigning low importance to highly predictable words regardless of their true importance.

For this reason, we choose to use a generative language model to infill masked sections with n-grams. Our mask-and-infill approach is similar to that of Wu et al. (2021) and Ross et al. (2021), who used fine-tuned causal language models to infill masked sections of text with variable length sequences. Ross et al. also used the contrasting label to condition the generative model. However, both these works aim to find counterfactual examples as explanations, while we marginalize over them to calculate necessity and sufficiency of each token.

3 Our Method

A central idea in causal inference is that of *intervention*, where a random variable is intervened on and set to a certain value. The intuition is that, if a random variable is the cause of another, then intervening on the first one should affect the other, whereas if they are correlated by other means then the intervention should not have an effect.

Necessity. Let $X \leftarrow a$ denote that the random variable X has been intervened so that $X = a$. When talking about a feature vector x, we will denote by $x_{i \leftarrow a}$ that we intervene on the ith feature value and set it to a. For an input with features x where $x_i = a$, the necessity of $x_i = a$ for the

[2]The term *comprehensiveness* is often used instead of *necessity* in this context.

[3]Making the softmax scores more distributed across the vocabulary results in unpredictably disfluent infills.

model prediction $f(x) = y$ is defined as:

$$N_{x_i,y}{=}P_{c\sim\mathcal{D}_n(x)}(f(c_{i\leftarrow a'}){=}y'|c_i = a, f(c) = y)$$

where a' is an alternative feature value such that $a' \neq a$ and y' is an alternate outcome such that $y' \neq y$. $\mathcal{D}_n(x)$ is a distribution that covers the neighborhood of x, and can be defined according to the data and the implementation. In words, $x_i = a$ has high necessity for the prediction y if, for those points in the neighborhood of x that also have the value a for the ith feature and the same model prediction y, changing the ith feature value from a to a' changes the prediction from y to y' with high probability.

Sufficiency. The sufficiency of $x_i = a$ for the model prediction $f(x) = y$ is defined as:

$$S_{x_i,y}{=}P_{c\sim\mathcal{D}_s(x)}(f(c_{i\leftarrow a}) = y|c_i = a', f(c) = y')$$

This means that if $x_i = a$ has high sufficiency for the outcome y, then for inputs in the neighborhood of x that differ in the ith feature value, changing ith feature value to that of a will flip the prediction to $f(x) = y$.

Interventions. Previous works applying notions of necessity and sufficiency from causal inference to XAI assume tabular data. This makes it relatively straightforward to apply these measures to the features since *a)* it is clear how to assess and compare the ith feature of each input and *b)* there is little ambiguity in how to change one feature value to another. Both these are issues for NLP data, where each feature is a token in the context of the wider text.

We argue that the replacements should reflect the likelihood of natural data, but should still be distinct from purely observational correlations in task-specific aspects. To achieve this balance, we sample the replacement values a' conditioned both on the other parts of the text and on the opposite class y'. If there are two features $x_i = a$ and $x_j = b$ that are both correlated with the outcome y, the intervention $x_{i\leftarrow a'}$, where a' is sampled in this way results in a' being still plausible with respect to the context $x_j = b$, but removes the potential indirect effect that $x_j = b$ causes $x_i = a$, which causes $f(x) = y$. This allows us to distinguish which of the correlated features the model relies on more for a given prediction.

Estimation. The formulae for necessity and sufficiency suggest a naive implementation of sampling first from the neighborhood of the input, picking those samples that conform to the conditions, and intervening on the feature of interest and marginalizing over the model predictions to calculate the final value. To perform these steps for each token in a sentence is prohibitively expensive. We therefore perform interventions on subsets of tokens at once, so that one perturbation can be used in the necessity and sufficiency estimation of multiple tokens.

We estimate the necessity of a token by perturbing subsets of tokens containing the given token and calculating the average change in model prediction, weighted according to the size of the subset. For calculating necessity, we marginalize over $f(c_{i\leftarrow a'})$ where $c = x_{j_1\leftarrow b_1,\cdots j_k\leftarrow b_k}$ for a random subset of features $x_{j_1}, \cdots, x_{j_k}$, not including the original feature i. This means that in our implementation, $\mathcal{D}_n(x)$ is an interventional distribution around x rather than an observational one. In practice, we estimate a simplified version of this value where we do not explicitly condition on $f(c) = y$ in order to perform the estimation efficiently.

We consider the instances where only one or a few tokens are perturbed to have higher probability in $\mathcal{D}_n(x)$. As such, the weight assigned to a sample with k perturbed tokens is proportional to $1/k$. This means that the difference between the original and the perturbed instance is attributed to each perturbed token equally.

For estimating sufficiency we take the dual approach. We perturb subsets of tokens *excluding* the target token, and calculate the difference between the weighted average of the model predictions and the baseline prediction. Here too, $\mathcal{D}_s(x)$ is an interventional distribution where each sample $c = x_{i\leftarrow a',j_1\leftarrow b_1,\cdots j_k\leftarrow b_k}$ for the focus feature x_i and a subset of other features $x_{j_1}, \cdots, x_{j_k}$. Even though we do not explicitly condition on $f(c) = y'$, $\mathcal{D}_s(x)$ is biased towards such c because the interventions are conditioned on y'. For a sequence of length n, the weight assigned to an instance where k tokens are perturbed is $1/(n-k)$. This means that for an perturbed example that contains only a single token from the original instance, the difference from the baseline will be attributed entirely to that token, whereas if there is k original tokens, the attribution is shared between them. Note that $\mathcal{D}_s(x)$ still assigns a higher probability mass to instances closer to x, but is less peaked than $\mathcal{D}_n(x)$.

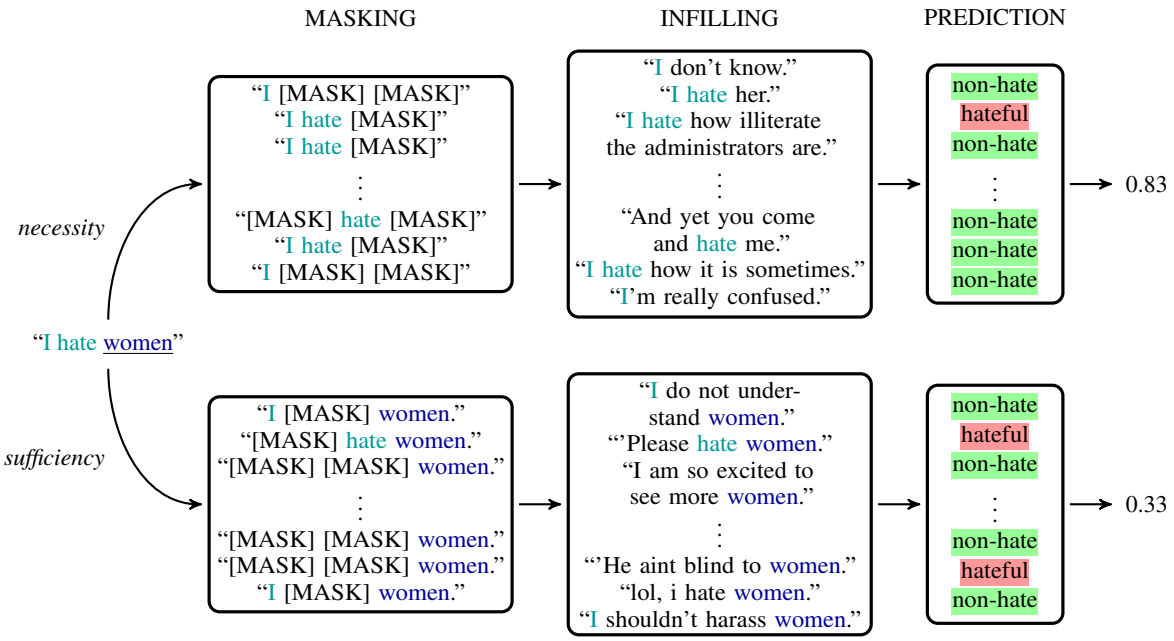

Figure 1: An illustration of how necessity and sufficiency are calculated for a chosen token "women" in the input "I hate women" that the model classifies as hateful. In the MASKING step, the subsets of tokens are masked. For the necessity calculation the masked tokens always include the focus word, and for sufficiency they always exclude it. In the INFILLING step, the generative language model is used to infill the masked sections with n-grams of various lengths. These are then passed to the classifier. The necessity is the proportion of instances where changing the token changes the prediction, and sufficiency is the proportion of instances where changing other tokens does not change the original prediction. The infills are real examples generated by our method.

4 Explaining Hate Speech Models

For tasks with very skewed class distributions such as those for binary hate speech classification, it is not intuitive to ask for explanations for the majority class predictions: it is difficult to answer why a regular utterance such as "I would like some coffee." is *not* hate speech. This echoes the argument of Miller (2019) that humans demand explanations only for selective and surprising aspects of an occurrence. We assume that there exists a majority, 'neutral' class, and aim to provide explanations only for the minority, 'positive' class.

At the core of our approach is sampling replacements $x_i \leftarrow a'$ that are interventional with respect to the task, but also have high likelihood in the context of other tokens. We implement such a perturbation model for explaining positive predictions of hate speech classifiers, and explore what information necessity and sufficiency provide for this task through quantitative and qualitative analysis. Our implementation is *task specific*, since we sample from the data labelled as 'neutral' only; however, it is not *dataset specific* in that it allows comparing hate speech classifiers that are trained on different datasets. Figure 1 illustrates the calculations of necessity and sufficiency for an example classifier.

4.1 Generating perturbations

We use the infilling language model (ILM) of Donahue et al. (2020) to generate the perturbed examples. This model fine-tunes GPT-2 (Radford et al., 2019) to allow infilling masked sections of text with n-grams, where the length of the infill varies from 1 to 7 tokens.

We fine-tune the ILM on training data that is labelled 'neutral', which is the opposite of the model prediction we aim to explain. Training the perturbation model only on the neutral examples allows us to distinguish direct causes of the model prediction from correlations in data. To see this, consider the case where we would like to determine whether a given identity mention is sufficient for a positive prediction. If the ILM captures the entire data distribution rather than just the neutral class, then a large chunk of the perturbed instances might still be hateful utterances. In those cases a good model should predict the positive class even though the occurrence of the identity term is arguably not the direct cause of the prediction. Rather, the presence of the identity term causes the ILM to generate a hateful infill, which then causes the positive pre-

	Functionality	Example Test Case	Gold Label	n
Explicit	**F1**: Expression of strong negative emotions	"I hate [IDENTITY]." "You [IDENTITY] make me sick."	hateful	40
	F2: Description using very negative attributes	"[IDENTITY] are disgusting." "All [IDENTITY] like them are vile."	hateful	40
	F3: Dehumanisation	"[IDENTITY] are scum." "[IDENTITY] are a pest."	hateful	40
Non-hate grp. ident.	**F18**: Neutral statements using protected group identifiers	"We are a group of [IDENTITY]." "I live with two [IDENTITY]."	non-hate	36
	F19: Positive statements using protected group identifiers	"I love [IDENTITY]." "[IDENTITY] are great."	non-hate	54
Abuse against non-prot. targets	**F22**: Abuse targeted at objects	"I really can't stand cauliflower." "All cocktails like these are vile"	non-hate	65
	F23: Abuse targeted at individuals (not as member of a prot. group)	"You make me sick." "I will hurt you."	non-hate	65
	F24: Abuse targeted at non-protected groups (e.g. professions)	"Artists are parasites to our society." "Death to all capitalists."	non-hate	62

Table 1: Test cases of interest from HateCheck (Röttger et al., 2021). We hypothesize that for identity terms in explicit test cases, if sufficiency is high then the model will have a high error rate on the test cases that capture *non-hate group identity mentions*, and if necessity is low then the model will have a high error rate on the test cases that cover *abuse against non-protected targets*.

diction. Using the non-hateful data distribution to train the infilling model helps avoid such cases, and enables the method to attribute importance to a token only when the classifier relies on it directly.

To train our infilling language model, we choose four widely-used datasets for hate speech and abusive language detection that are from various sources such as Twitter (Founta et al., 2018), Reddit (Vidgen et al., 2021), Wikipedia comments (Wulczyn et al., 2017) and news article comments (Borkan et al., 2019). For all these datasets, we fine-tune our generative model on benign instances from the training sets. Details for training and datasets can be found in Appendix A.

5 Experiments

For our experiments, we focus on investigating the necessity and sufficiency of identity mentions for a sentence to be detected as hate speech, since a known bias in many hate speech detection models is that mere mentions of identity terms result in false positive predictions (Dixon et al., 2018).

For a set of instances that contain mentions of identity terms we leverage HateCheck (Röttger et al., 2021), which is a suite of functional tests targeted at diagnosing weaknesses of hate speech classifiers. Tests are constructed from hand-crafted templates, where the target is picked from a predefined set of protected groups.

To evaluate our explanation method, we train six BERT classifiers on three different datasets (Founta et al., 2018; Vidgen et al., 2021; Davidson et al., 2017), and obtain the necessity and sufficiency of the identity terms on explicitly hateful test cases in HateCheck that target *women* and *Muslims*. Details for the datasets and classifiers can be found in Appendix B. We train classifiers on both 'hate speech' labels and on more general 'abusive' language labels in order to observe the effects of this distinction on the necessity and sufficiency values for identity terms. The difference between the two is that abusive language does not need to target a protected group identity. Accordingly, our first hypothesis is:

Hypothesis 1: *We will observe lower necessity for the identity terms for those models that are trained on the 'abuse' labels compared to the ones that are trained on the 'hate' labels.*

We further hypothesize that different necessity and sufficiency scores for identity mentions in explicitly hateful test cases indicate different biases, and correlate with how well a model does in the HateCheck functionalities that aim to capture those biases (see Table 1 for the test cases), leading to our

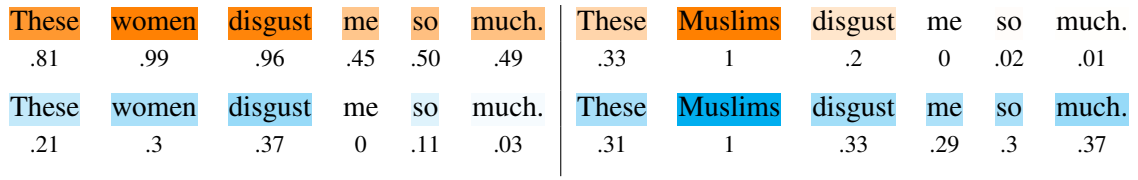

Figure 2: Visualization for necessity and sufficiency scores on an example HateCheck test case targeting *women* and *Muslims* for the classifier *Founta2018-hate*. Darker shades correspond to higher values.

second and third hypotheses:

Hypothesis 2: *If a model has high sufficiency scores for identity terms in explicit expressions of hate (functionalities **F1**, **F2** and **F3**), this should indicate that the model is over-sensitive to identity terms. Therefore, we expect it have increased error rate in **F18** and **F19**, which consist of neutral or positive statements that mention identity terms.*

Hypothesis 3: *If the necessity scores for identity terms are low in explicit expressions of hate, we can conclude under-reliance on the identity terms, and over-reliance on the context. Consequently, we expect that the model will perform poorly on **F22**, **F23** and **F24**, which capture abuse not targeted at protected identity groups.*

5.1 Implementation

We obtain the average necessity and sufficiency values for explicitly hateful test cases targeting *women* and *Muslims* for each of the classifiers. We calculate necessity and sufficiency by masking a subset of the tokens and using our fine-tuned language model to generate infillings. If multiple consecutive tokens are chosen, we aggregate them to a single mask instance to be infilled. We choose the number of perturbations for each example so that the expected number of perturbations for each token is 100. The necessity and sufficiency scores are only calculated for test cases that a classifier returns a correct prediction, since we only aim to explain positive predictions. The results can be found in Table 2. Table 3 presents the proportions of test cases classified as hateful/abusive by each of the six classifiers on the non-hateful statements that mention identity terms (**F18** and **F19**) and abusive utterances not targeting protected identity groups (**F22**, **F23**, and **F24**). We report the results where necessity and sufficiency are calculated with masking rather than perturbing the chosen tokens in Appendix C.

As baselines, we calculate the average importance of the tokens corresponding to target groups with SHAP[4] and LIME[5]. For both of these methods, we use the default parameters for textual data. As with the calculation of necessity and sufficiency, we only include the attribution scores for test cases on which the classifier correctly predicts the positive class. These results can be found in Table 4.

6 Results and Discussion

An example necessity and sufficiency attribution is given in Figure 2. It shows that for this input, the token 'Muslims' is more sufficient compared to 'women', and the token 'disgust' is more necessary in the context of 'women' than that of 'Muslims'.

According to our first hypothesis, we expect the models that were trained on the *abuse* versions of each dataset to have lower necessity for identity terms compared to those that have been trained on *hate* labels. Indeed, in Table 2 we observe this pattern for all models and targets except *Davidson2017* for the target *women*. This correctly suggests that identity terms are necessary for a comment to be hate speech, but not for it to be abusive.

The results also clearly support our second hypothesis that if an identity mention has high sufficiency on explicit examples for a given model, then this model is over-sensitive to the identity term. Comparing the sufficiency of *women* and *Muslims* in Table 2 illustrates this difference: for all models except *Davidson2017-abuse* sufficiency is high for *Muslims* and significantly lower for *women*. Accordingly, all models except *Davidson2017-abuse* display a large difference between their error rates on neutral or positive mentions for *women* and *Muslims* in Table 3 (**F18**, **F19**). That is, the mere occurrence of the word "Muslims" is sufficient for the classifiers to classify a text as hate speech, even if the text is neutral. Furthermore within each group, higher sufficiency values correspond to higher error rates in functionalities **F18, F19**. *Vidgen2021-hate*

[4] https://github.com/slundberg/shap
[5] https://github.com/marcotcr/lime

	women		**Muslims**	
	necc	*suff*	*necc*	*suff*
Founta2018-hate	0.82 ±0.18	0.29 ±0.1	0.89 ±0.16	0.81 ±0.08
Founta2018-abuse	0.54 ±0.17	0.34 ±0.1	0.65 ±0.21	0.82 ±0.06
Davidson2017-hate	0.58 ±0.09	0.21 ±0.06	0.91 ±0.12	0.74 ±0.09
Davidson2017-abuse	0.82 ±0.14	0.43 ±0.13	0.83 ±0.13	0.41 ±0.14
Vidgen2021-hate	0.96 ±0.02	0.71 ±0.17	0.97 ±0.03	0.88 ±0.13
Vidgen2021-abuse	0.82 ±0.14	0.64 ±0.14	0.82 ±0.15	0.88 ±0.07

Table 2: The mean and standard deviation of necessity and sufficiency scores for target tokens in explicitly hateful cases of HateCheck (**F1**, **F2**, and **F3**) targeting **women** or **Muslims** for the the three classifiers trained on hate, and three classifiers trained on abuse labels.

	Neutral/supportive group identity mention **(F18, F19)**		*Abuse against non-protected targets* **(F22, F23, F24)**		
	women	**Muslims**	**group**	**individual**	**object**
Founta2018-hate	0.02	0.78	0.19	0.15	0.05
Founta2018-abuse	0.02	0.78	0.45	0.72	0.37
Davidson2017-hate	0.02	0.78	0.37	0.18	0.02
Davidson2017-abuse	0.31	0.22	0.26	0.28	0.14
Vidgen2021-hate	0.36	0.82	0.02	0.00	0.00
Vidgen2021-abuse	0.42	0.96	0.40	0.61	0.00

Table 3: Proportions of test cases classified as hateful/abusive for different non-hateful HateCheck functionalities and targets.

and *Vidgen2021-abuse* display the highest sufficiency for *women*, and correspondingly have the highest error rates on these test cases for *women*. *Davidson2017-abuse* has the lowest sufficiency for *Muslims*, and the lowest error rate for this target.

Our third hypothesis is that low necessity for identity terms will be correlated with positive predictions for abusive instances that do not target a protected identity. In Table 2, the lowest necessity for both target groups are observed with *Founta2018-abuse*. Indeed, this model has the highest rate of positive (abuse) predictions on all functionalities that test for abuse against non-protected targets in Table 3. The false positives in the test cases that target *objects* is much higher than the corresponding errors for the other models, indicating that *Founta2018-abuse* is indeed over-sensitive to abusive contexts, and does not consider the target of the abuse to be a necessary feature for the classification. On the other hand, the classifier trained on *Vidgen2021-hate* shows the highest necessity values for both targets, and the lowest error rates on **F22**, **F23**, **F24**.

6.1 Comparison of Average SHAP and LIME Values with Necessity and Sufficiency

The average SHAP and LIME values for the two targets are presented in Table 4. While *Founta-abuse* and *Davidson-abuse* get very similar SHAP scores for the target *Muslims*, *Founta2018-abuse* has high sufficiency for this token while *Davidson2017-abuse* has high necessity. These two classifiers have very different false-positive rates for test instances that are non-abusive mentions of this target as reported in Table 3, and hence can be observed to be biased against this group to a different extent. This distinction is clearly captured with the necessity and sufficiency scores, but not with SHAP.

LIME scores seem even less consistent with the false-positive rates in Table 3 than SHAP. For example, *Davidson2017-hate* has a very high false-positive rate for neutral/supportive mentions of the target *Muslims*, however the average LIME score for this model and target group is in low negatives. This means that LIME was unable to capture the biases of the model against this target group.

For the target *women*, LIME gives very similar average importance to the target tokens for *Founta2018-abuse* and *Vidgen2021-hate*, however all of the other metrics we calculate highlight significant differences. Indeed, we can observe in Table 3 that *Founta2018-abuse* has high false-positive rates for abuse against non-protected targets, but low false-positive rates for neutral/supportive mentions of the target *women*, whereas *Vidgen2021-hate* shows the exact opposite pattern. LIME scores seem unable to distinguish this important difference, unlike our necessity and sufficiency scores.

	women		Muslims	
	SHAP	*LIME*	*SHAP*	*LIME*
Founta2018-hate	0.36 ±0.24	0.018 ±0.045	0.84 ±0.2	0.013 ±0.036
Founta2018-abuse	-0.07 ±0.21	0.011 ±0.02	0.39 ±0.25	0.042 ±0.037
Davidson2017-hate	0.01 ±0.02	-0.021 ±0.018	0.89 ±0.15	-0.002 ±0.057
Davidson2017-abuse	0.42 ±0.19	0.001 ±0.086	0.37 ±0.2	-0.032 ±0.092
Vidgen2021-hate	0.89 ±0.17	0.012 ±0.077	0.95 ±0.11	0.042 ±0.049
Vidgen2021-abuse	0.66 ±0.23	0.045 ±0.088	0.75 ±0.22	0.087 ±0.062

Table 4: The mean and standard deviation of SHAP and LIME scores for target tokens in explicitly hateful cases of HateCheck (**F1**, **F2**, and **F3**) targeting **women** or **Muslims** for the the three classifiers trained on hate, and three classifiers trained on abuse labels.

6.2 Qualitative analysis

Examining the generated perturbations reveals that the majority of the time, the generator successfully replaces the original snippet with non-offensive text. There are minimal repetitions of the same infill, indicating that the perturbations are sufficiently diverse. The quality of the perturbations is mixed, and tends to decrease with the number of infills. Our method can be used with any generator that can model the data distribution conditioned on the label. For our current model we prioritize diversity over fluency, and leave improvements on this aspect for future work.

Inspecting the generated perturbations reveal that there are often instances that combine a few HateCheck functionalities. For example, as a perturbation for "I hate [IDENTITY]" model generates "This is a subreddit I hate.", combining shuffling the word order with changing the target. For the template "[IDENTITY] are the dumbest people.", one of the generated counterfactuals is "I don't believe that American people are the dumbest people.", combining negation and phrasing.

Interestingly, many infills for the identity terms do not clearly correspond to either object, individuals or non-protected groups, which are the three cases that HateCheck checks for. Instead, our generative model infills it with clauses such as "I hate that I feel like I have to do these things", actions such as "I hate crying," or types of events like "I hate surprises." This illustrates that our relatively non-constrained generation of counterfactuals provides better coverage of potential replacements, and provides a good complement to manual checks.

7 Conclusion

This work is a step towards more informative and transparent feature attribution metrics for explaining text classifiers. We argue that standard token importance metrics can be ambiguous in terms of what 'importance' means. Instead, we adapt the theoretically-grounded concepts of *necessity* and *sufficiency* to explain text classifiers. Besides being more informative, the process of generating these two metrics is intuitive and can be explained to lay people in terms of "how much the perturbations in input change the output of the classifier". Moreover, the input perturbations can be presented to the users, leading to a transparent and understandable explainability framework.

Considering the complexities of perturbing textual features, we introduced a practical implementation to compute the necessity and sufficiency of the input tokens. Taking hate speech detection as an example application, we showed that sufficiency and necessity can be used to explain the expected differences between a classifier that is intended to detect identity-based hate speech and those trained for detecting general abuse. We also leveraged these metrics to explain the observed over-sensitivity and under-sensitivity to mentions of target groups, issues that are tightly related to fairness in hate speech detection. While the current work focused on binary hate speech detection for English-language social media posts, in future work, we will explore the effectiveness of these metrics in generating explanations for other applications and languages. We will also explore how the new metrics can improve the debugging of the models or communicating the model's decision-making process to the end-users.

8 Ethical Considerations

The proposed method has benefits and risks that should be considered from an ethics perspective.

One principle of ethical AI is *transparency*, and we have developed this method with the goal of improving transparency for system developers, end users, and other stakeholders to better understand

the inner workings of complex NLP systems. In the application domain of hate speech detection, we demonstrated how necessity and sufficiency scores might be used to diagnose possible classification biases against identity groups, who are frequently subjects of online abuse. This can help in addressing the known issue of over-sensitivity to identity terms, ensuring that benign conversations around issues concerning marginalized groups are not misclassified as hate speech.

However, there are also potential risks. We make use of existing datasets and thus our analysis is limited by those data: they were collected from public, online platforms without user's explicit consent, and may not accurately represent speakers from all demographic groups, they are only in English, and they may be biased towards or against certain topics of conversation. The data and analysis are also limited to the English language. Training language models on user data also has privacy implications, as the language model may then re-generate user text when deployed.

While transparency and explainability are seen as desirable properties, they can also expose AI systems to malicious attacks. In the context of hate speech, our explainability metrics could potentially be used to identify and then exploit system vulnerabilities.

Finally, our approach requires the use of large language models, which are computationally expensive to train and can reflect the biases of their training data. Our method of generating multiple counterfactual examples per word, rather than simply removing or masking that word, also increases the computational resources required.

References

Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6:52138–52160.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion Proceedings of the 2019 World Wide Web Conference*, pages 491–500.

Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. 2018. Explaining image classifiers by counterfactual generation. In *Proceedings of the International Conference on Learning Representations*.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable AI for natural language processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459, Suzhou, China. Association for Computational Linguistics.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11.

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 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2020. Eraser: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458.

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*, pages 67–73.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2492–2501.

Ruth C Fong and Andrea Vedaldi. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pages 3429–3437.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30.

Antigoni Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of Twitter abusive behavior. In *Proceedings of the Twelfth International AAAI Conference on Web and Social Media*.

Sainyam Galhotra, Romila Pradhan, and Babak Salimi. 2021. Explaining black-box algorithms using probabilistic contrastive counterfactuals. In *Proceedings of the 2021 International Conference on Management of Data*, pages 577–590.

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.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42.

Joseph Y. Halpern. 2016. *Actual Causality*. MIT Press.

David Harbecke and Christoph Alt. 2020. Considering likelihood in NLP classification explanations with occlusion and language modeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 111–117.

Peter Hase, Harry Xie, and Mohit Bansal. 2021. The out-of-distribution problem in explainability and search methods for feature importance explanations. *Advances in Neural Information Processing Systems*, 34.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. 2019. A benchmark for interpretability methods in deep neural networks. *Advances in Neural Information Processing Systems*, 32:9737–9748.

Alon Jacovi and Yoav Goldberg. 2021. Aligning faithful interpretations with their social attribution. *Transactions of the Association for Computational Linguistics*, 9:294–310.

Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. 2020. Feature relevance quantification in explainable AI: A causal problem. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 2907–2916. PMLR.

Divyansh Kaushik, Amrith Setlur, Eduard H Hovy, and Zachary Chase Lipton. 2021. Explaining the efficacy of counterfactually augmented data. In *Proceedings of the International Conference on Learning Representations*.

Siwon Kim, Jihun Yi, Eunji Kim, and Sungroh Yoon. 2020. Interpretation of NLP models through input marginalization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3154–3167, Online. Association for Computational Linguistics.

Svetlana Kiritchenko, Isar Nejadgholi, and Kathleen C Fraser. 2021. Confronting abusive language online: A survey from the ethical and human rights perspective. *Journal of Artificial Intelligence Research*, 71:431–478.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc.

Siwen Luo, Hamish Ivison, Caren Han, and Josiah Poon. 2021. Local interpretations for explainable natural language processing: A survey. *arXiv preprint arXiv:2103.11072*.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14867–14875.

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38.

Ramaravind Mothilal, Divyat Mahajan, Chenhao Tan, and Amit Sharma. 2021. Towards unifying feature attribution and counterfactual explanations: Different means to the same end. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 652–663.

Isar Nejadgholi and Svetlana Kiritchenko. 2020. On cross-dataset generalization in automatic detection of online abuse. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 173–183.

Ji Ho Park, Jamin Shin, and Pascale Fung. 2018. Reducing gender bias in abusive language detection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2799–2804, Brussels, Belgium.

Judea Pearl. 2009. *Causality*. Cambridge University Press.

Thang M Pham, Trung Bui, Long Mai, and Anh Nguyen. 2021. Double trouble: How to not explain a text classifier's decisions using counterfactuals synthesized by masked language models? *arXiv preprint arXiv:2110.11929*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

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

Alexis Ross, Ana Marasović, and Matthew E Peters. 2020. Explaining NLP models via minimal contrastive editing (MiCE). *arXiv preprint arXiv:2012.13985*.

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.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the International Conference on Machine Learning*, pages 3319–3328. PMLR.

Jennifer Wortman Vaughan and Hanna Wallach. 2021. A human-centered agenda for intelligible machine learning. In Marcello Pelillo and Teresa Scantamburlo, editors, *Machines We Trust: Perspectives on Dependable AI*. The MIT Press.

Bertie Vidgen, Dong Nguyen, Helen Margetts, Patricia Rossini, and Rebekah Tromble. 2021. Introducing CAD: the contextual abuse dataset. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2289–2303.

David Watson, Limor Gultchin, Ankur Taly, and Luciano Floridi. 2021. Local explanations via necessity and sufficiency: Unifying theory and practice. In *Proceedings of Machine Learning Research*, volume 161, page 1382–1392.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel 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, Online. Association for Computational Linguistics.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1391–1399.

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 260–267.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. 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 1298–1308.

A Data, Training and Generation Details for the Infilling Language Model

To fine-tune the ILM model, we use the following four datasets: *Wikipedia Toxicity*[6] (Wulczyn et al., 2017), *Founta2018*[7] (Founta et al., 2018), *Civil Comments*[8] (Borkan et al., 2019), and *Vidgen2021*[9] (Vidgen et al., 2021). The datasets contain English-language utterances, and cover different domains (Twitter post, Reddit posts, Wikipedia comments, and comments from news websites). The datasets have been created to study abusive language, and are commonly used to train and evaluate classification models that detect various sub-categories of online abuse, such as hate speech, toxicity, personal attacks, etc. All datasets except *Founta2018* are in the public domain and licensed for research purposes. *Founta2018* dataset is being used with the permission of the first author.

The details on each dataset are provided in Table A.1. For the *Wikipedia Toxicity* dataset, a large portion of the data is from conversations about Wikipedia-specific topics. To not skew our generation model, we filter these instances following the unsupervised method presented by Nejadgholi and Kiritchenko (2020)[10]. Because the *Civil Comments* dataset is significantly larger than the rest, we randomly sample 30K neutral instances and discard the rest. After filtering, the compound dataset of neutral instances consists of 130,430 instances in total. As preprocessing, we replace URLs, mentions and emojis with special tokens.

To train the ILM, we fine-tune GPT-2 (1.5B parameters) for 4 epochs with the default hyperparameters provided by Donahue et al. (2020). The training takes approximately 2.5 hours on a Tesla V100-SXM2 GPU. Although the original ILM is trained by infilling words, n-grams, sentences and paragraphs, we modify the objective to only infill words and n-grams.

We generate perturbations once for the 120 HateCheck cases, and evaluate all models on the same set of perturbations. The number of perturbations are chosen so that to have approximately 100 perturbed instances for each token for the necessity calculation, and 100 instances for the sufficiency calculation. This results in a total of 66,120 perturbed instances, and takes approximately 6 hours to generate on a 2.3 GHz Quad-Core Intel Core i7 CPU.

Dataset	Source	Class	Size
Wikipedia Toxicity (Wulczyn et al., 2017)	Wikipedia comments	Normal	36,121
Founta2018 (Founta et al., 2018)	Twitter posts	Normal	53,236
Civil Comments (Borkan et al., 2019)	Comments on news sites	Normal	30,000
Vidgen2021 (Vidgen et al., 2021)	Reddit posts	Non-Abusive	11,073
Total			**130,430**

Table A.1: Description of the training data used to fine-tune the ILM model.

B Data and Training Details for Hate Speech Classifiers

We fine-tune six BERT (Devlin et al., 2019) classifiers on three different datasets and with two different labelling schemes (hate speech vs. abusive language) for each. The datasets include: *Davidson2017*[11] (Davidson et al., 2017), *Founta2018* (Founta et al., 2018), and *Vidgen2021* (Vidgen et al., 2021). The datasets contain English-language posts from two online platforms, Twitter and Reddit. The details on each dataset are provided in Table B.1.

We train two models on the dataset of Founta et al. (2018). For *Founta2018-hate*, we binarize the labels to map hate annotations as positive, and the rest as the negative class. For *Founta2018-abuse*, we label both hate and abuse annotations as positive, and the rest as negative. To illustrate that our method can provide explanations for models trained on data that is not explicitly modelled by our perturbation generator, we also train models on two versions of the dataset of Davidson et al. (2017): *Davidson2017-abuse* and *Davidson2017-hate*, which are binarized in the same manner.

The dataset of Vidgen et al. (2021) provides a hierarchical labelling scheme, the top distinction being *abusive* vs. *non-abusive*. We binarize *Vidgen2021-abuse* based on these labels. For *Vidgen2021-hate*, we take the positive class to be those instances that are labelled *identity-directed abuse*, and label the rest as the negative class.

[6] https://figshare.com/articles/dataset/Wikipedia_Talk_Labels_Toxicity/4563973

[7] https://github.com/ENCASEH2020/hatespeech-twitter

[8] https://bit.ly/3Kfaveb

[9] https://zenodo.org/record/4881008#.YeBBQ2jMKUk

[10] https://github.com/IsarNejad/cross_dataset_toxicity

[11] https://github.com/t-davidson/hate-speech-and-offensive-language/tree/master/data

Classifier	Dataset	Positive Class	Negative Class	Size		
				Train	*Dev*	*Test*
Founta2018-hate	Founta2018 (Founta et al., 2018)	Hateful	Normal Abusive	62,445	7,806	7,806
Founta2018-abuse	Founta2018 (Founta et al., 2018)	Hateful Abusive	Normal	62,445	7,806	7,806
Davidson2017-hate	Davidson2017 (Davidson et al., 2017)	Hate	Neither Offensive	19,826	2,478	2,479
Davidson2017-abuse	Davidson2017 (Davidson et al., 2017)	Hate Offensive	Neither	19,826	2,478	2,479
Vidgen2021-hate	Vidgen2021 (Vidgen et al., 2021)	Identity-directed abuse	Non-abusive Person-directed abuse Affiliation-directed abuse	13,585	4,527	5,308
Vidgen2021-abuse	Vidgen2021 (Vidgen et al., 2021)	Abusive	Non-abusive	13,585	4,527	5,308

Table B.1: Description of the datasets used to fine-tune hate speech and abuse detection classifiers.

	Micro F1	Macro F1	Training time (mins)
Founta2018-hate	0.94	0.67	28
Founta2018-abuse	0.94	0.93	28
Davidson2017-hate	0.94	0.70	7
Davidson2017-abuse	0.96	0.93	7
Vidgen2021-hate	0.91	0.71	22
Vidgen2021-abuse	0.85	0.72	22

Table B.2: Micro- and macro-averaged F1-scores and training times for each BERT model trained and evaluated on the given datasets.

	women		**Muslims**	
	necc	*suff*	*necc*	*suff*
Founta2018-hate	0.53	0.30	0.72	0.81
Founta2018-abuse	0.19	0.34	0.36	0.82
Davidson2017-hate	0.44	0.21	0.88	0.74
Davidson2017-abuse	0.55	0.44	0.52	0.41
Vidgen2021-hate	0.87	0.71	0.93	0.88
Vidgen2021-abuse	0.62	0.64	0.64	0.88

Table B.3: Average necessity and sufficiency scores calculated by masking rather than perturbing selected tokens, for the identity terms in explicitly hateful cases of HateCheck (**F1**, **F2**, and **F3**) targeting **women** or **Muslims** for the the three classifiers trained on hate, and three classifiers trained on abuse labels.

We employ the same pre-processing steps as in the experiments by Röttger et al. (2021), and replace URLs, mentions and emojis with special tokens. We fine-tune a BERT model from the Hugging Face library[12] on each of these datasets on a single Tesla V100-SXM2 GPU. Each model has 110M trainable parameters. We follow the implementation of Röttger et al. (2021) and use their hyper-parameters of 3 epochs, batch size of 16, learning rate of 5e-5 and weight decay of 0.01. We also employ weighted cross-entropy loss that corrects for the class imbalance in data. For the training/development/test splits, we use the standard split for *Vidgen2021* provided by the creators of the dataset, and use a stratified 80/10/10 split for the other datasets, making sure that the splits are the same for the *hate* and *abuse* versions of each, and correspond to the training set for ILM when applicable. The classification performance of these models on the held-out test sets is shown in Table B.2, together with the training times for each. We can observe that the reported scores are within a few percentage points of the previously published results (Röttger et al., 2021). All reported results are from a single run.

[12] https://huggingface.co/bert-base-uncased

C Calculating Necessity and Sufficiency with Masking

In Section 1 we have argued that using the *mask* token from the pre-training objective in feature attribution methods has several drawbacks. Nevertheless, in Table B.3 we report the results of a modified version of our experiment presented in Section 5 where we keep the number and the location of the perturbations the same as the original experiments, but instead of perturbing the chosen tokens using an LM, we replace them with the *mask* token. The results show that although the values are different than their counterparts in the main experiment, the overall trends remain the same, and support the hypotheses presented in Section 5. Evaluating the classifier with the masked input is faster than explicitly generating perturbations, but the method ceases to be model agnostic and looses transparency. The results still suggest that

evaluating necessity and sufficiency with masked rather than perturbed inputs might be preferable in contexts where latency is more important than transparency, or as a pre-processing step to choose which inputs and tokens to focus on for in-depth analysis with explicit perturbations. We leave further explorations of this avenue for future work.