Incorporating Priors with Feature Attribution on Text Classification

Frederick Liu **Besim Avci**
Google
{fredericklliu, besim}@google.com

Abstract

Feature attribution methods, proposed recently, help users interpret the predictions of complex models. Our approach integrates feature attributions into the objective function to allow machine learning practitioners to incorporate priors in model building. To demonstrate the effectiveness our technique, we apply it to two tasks: (1) mitigating unintended bias in text classifiers by neutralizing identity terms; (2) improving classifier performance in a scarce data setting by forcing the model to focus on toxic terms. Our approach adds an L_2 distance loss between feature attributions and task-specific prior values to the objective. Our experiments show that i) a classifier trained with our technique reduces undesired model biases without a tradeoff on the original task; ii) incorporating priors helps model performance in scarce data settings.

1 Introduction

One of the recent challenges in machine learning (ML) is interpreting the predictions made by models, especially deep neural networks. Understanding models is not only beneficial, but necessary for wide-spread adoption of more complex (and potentially more accurate) ML models. From healthcare to financial domains, regulatory agencies mandate entities to provide explanations for their decisions (Goodman and Flaxman, 2016). Hence, most machine learning progress made in those areas is hindered by a lack of model explainability – causing practitioners to resort to simpler, potentially low-performance models. To supply for this demand, there has been many attempts for model interpretation in recent years for tree-based algorithms (Lundberg et al., 2018) and deep learning algorithms (Lundberg and Lee, 2017; Smilkov et al., 2017; Sundararajan et al., 2017; Bach et al., 2015; Kim et al., 2018; Dhurandhar et al., 2018).

Method	Sentence	Probability
Baseline	I am *gay*	0.915
	I am straight	0.085
Our Method	*I* *am* gay	0.141
	I *am* straight	0.144

Table 1: Toxicity probabilities for samples of a baseline CNN model and our proposed method. Words are shaded based on their attribution and italicized if attribution is > 0.

On the other hand, the amount of research focusing on explainable natural language processing (NLP) models (Li et al., 2016; Murdoch et al., 2018; Lei et al., 2016) is modest as opposed to image explanation techniques.

Inherent problems in data emerge in a trained model in several ways. Model explanations can show that the model is not inline with human judgment or domain expertise. A canonical example is model unfairness, which stems from biases in the training data. Fairness in ML models rightfully came under heavy scrutiny in recent years (Zhang et al., 2018a; Dixon et al., 2018; Angwin et al., 2016). Some examples include sentiment analysis models weighing negatively for inputs containing identity terms such as "jew" and "black", and hate speech classifiers leaning to predict *any* sentence containing "islam" as toxic (Waseem and Hovy, 2016). If employed, explanation techniques help divulge these issues, but fail to offer a remedy. For instance, the sentence "I am gay" receives a high score on a toxicity model as seen in Table 1. The Integrated Gradients (Sundararajan et al., 2017) explanation method attributes the majority of this decision to the word "gay." However, none of the explanations methods suggest next steps to fix the issue. Instead, researchers try to reduce biases indirectly by mostly adding more data (Dixon et al.,

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6274–6283
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

2018; Chen et al., 2018), using unbiased word vectors (Park et al., 2018), or directly optimizing for a fairness proxy with adversarial training (Madras et al., 2018; Zhang et al., 2018a). These methods either offer to collect more data, which is costly in many cases, or make a tradeoff between original task performance and fairness.

In this paper, we attempt to enable injecting priors through model explanations to rectify issues in trained models. We demonstrate our approach on two problems in text classification settings: (1) model biases towards protected identity groups; (2) low classification performance due to lack of data. The core idea is to add L_2 distance between Path Integrated Gradients attributions for pre-selected tokens and a target attribution value in the objective function as a loss term. For model fairness, we impose the loss on keywords identifying protected groups with target attribution of 0, so the trained model is penalized for attributing model decisions to those keywords. Our main intuition is that undesirable correlations between toxicity labels and instances of identity terms cause the model to learn unfair biases which can be corrected by incorporating priors on these identity terms. Moreover, our approach allows practitioners to impose priors in the other direction to tackle the problem of training a classifier when there is only a small amount of data. As shown in our experiments, by setting a positive target attribution for known toxic words [1], one can improve the performance of a toxicity classifier in a scarce data regime.

We validate our approach on the Wikipedia toxic comments dataset (Wulczyn et al., 2017). Our fairness experiments show that the classifiers trained with our method achieve the same performance, if not better, on the original task, while improving AUC and fairness metrics on a synthetic, unbiased dataset. Models trained with our technique also show lower attributions to identity terms on average. Our technique produces much better word vectors as a by-product when compared to the baseline. Lastly, by setting an attribution target of 1 on toxic words, a classifier trained with our objective function achieves better performance when only a subset of the data is present.

[1]Full list of identity terms and toxic terms used as priors can be found in supplemental material. Please note the toxic terms are not censored.

2 Feature Attribution

In this section, we give formal definitions of feature attribution and a primer on [Path] Integrated Gradients (IG), which is the basis for our method.

Definition 2.1. Given a function $f : R^n \rightarrow [0, 1]$ that represents a model, and an input $\boldsymbol{x} = (x_1, ..., x_n) \in R^n$. An attribution of the prediction at input $\boldsymbol{x}$ is a vector $\boldsymbol{a} = (a_1, ..., a_n)$ and a_i is defined as the attribution of x_i.

Feature attribution methods have been studied to understand the contribution of each input feature to the output prediction score. This contribution, then, can further be used to interpret model decisions. Linear models are considered to be more desirable because of their implicit interpretability, where feature attribution is the product of the feature value and the coefficient. To some, non-linear models such as gradient boosting trees and neural networks are less favorable due to the fact that they do not enjoy such transparent contribution of each feature and are harder to interpret (Lou et al., 2012).

Despite the complexity of these models, prior work has been able to extract attributions with gradient based methods (Smilkov et al., 2017), Shapley values from game theory (SHAP) (Lundberg and Lee, 2017), or other similar methods (Bach et al., 2015; Shrikumar et al., 2017). Some of these attributions methods, for example Path Intergrated Gradients and SHAP, not only follow Definition 2.1, but also satisfy axioms or properties that resemble linear models. One of these axioms is completeness, which postulates that the sum of attributions should be equal to the difference between uncertainty and model output.

Integrated Gradients

Integrated Gradients (Sundararajan et al., 2017) is a model attribution technique applicable to all models that have differentiable inputs w.r.t. outputs. IG produces feature attributions relative to an uninformative baseline. This baseline input is designed to produce a high-entropy prediction representing uncertainty. IG, then, interpolates the baseline towards the actual input, with the prediction moving from uncertainty to certainty in the process. Building on the notion that the gradient of a function, f, with respect to input can characterize sensitivity of f for each input dimension, IG simply aggregates the gradients of f with respect to the input along this path using a path integral.

The crux of using path integral rather than overall gradient at the input is that f's gradients might have been saturated around the input and integrating over a path alleviates this phenomenon. Even though there can be infinitely many paths from a baseline to input point, *Integrated Gradients* takes the straight path between the two. We give the formal definition from the original paper in 2.2.

Definition 2.2. Given an input $\boldsymbol{x}$ and baseline $\boldsymbol{x'}$, the integrated gradient along the i^{th} dimension is defined as follows.

$$IG_i(\boldsymbol{x}, \boldsymbol{x'}) ::= (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial f(\boldsymbol{x'} + \alpha \times (\boldsymbol{x} - \boldsymbol{x'}))}{\partial x_i} \, d\alpha \quad (1)$$

where $\frac{\partial f(\boldsymbol{x})}{\partial x_i}$ represents the gradient of f along the i^{th} dimension at $\boldsymbol{x}$.

In the NLP setting, $\boldsymbol{x}$ is the concatenated embedding of the input sequence. The attribution of each token is the sum of the attributions of its embedding.

There are other explainability methods that attribute a model's decision to its features, but we chose IG in this framework due to several of its characteristics. First, it is both theoretically justified (Sundararajan et al., 2017) and proven to be effective in NLP-related tasks (Mudrakarta et al., 2018). Second, the IG formula in 2.2 is differentiable everywhere with respect to model parameters. Lastly, it is lightweight in terms of implementation and execution complexity.

3 Incorporating Priors

Problems in data manifest themselves in a trained model's performance on classification or fairness metrics. Traditionally, model deficiencies were addressed by providing priors through extensive feature engineering and collecting more data. Recently, attributions help uncover deficiencies causing models to perform poorly, but do not offer actionability.

To this end, we propose to add an extra term to the objective function to penalize the L_2 distance between model attributions on certain features and target attribution values. This modification allows model practitioners to inject priors. For example, consider a model that tends to predict every sentence containing "gay" as toxic in a comment moderation system. Penalizing non-zero attributions on the tokens identifying protected groups would force the model to focus more on the context words rather than mere existence of certain tokens.

We give the formal definition of the new objective function that incorporates priors as the follows:

Definition 3.1. Given a vector $\boldsymbol{t}$ of size n, where n is the length of the input sequence and t_i is the attribution target value for the ith token in the input sequence. The *prior* loss for a scalar output is defined as:

$$\mathcal{L}^{prior}(\boldsymbol{a}, \boldsymbol{t}) = \sum_{i}^{n} (a_i - t_i)^2 \quad (2)$$

where a_i refers to attribution of the ith token as in Definition 2.1.

For a multi-class problem, we train our model with the following joint objective,

$$\mathcal{L}^{joint} = \mathcal{L}(\boldsymbol{y}, \boldsymbol{p}) + \lambda \sum_{c}^{C} \mathcal{L}^{prior}(\boldsymbol{a^c}, \boldsymbol{t^c}) \quad (3)$$

where $\boldsymbol{a^c}$ and $\boldsymbol{t^c}$ are the attribution and attribution target for class c, λ is the hyperparameter that controls the stength of the prior loss and $\mathcal{L}$ is the cross-entropy loss defined as follows:

$$\mathcal{L}(\boldsymbol{y}, \boldsymbol{p}) = \sum_{c}^{C} -y_c \log(p_c) \quad (4)$$

where $\boldsymbol{y}$ is an indicator vector of the ground truth label and p_c is the posterior probability of class c.

The *joint* objective function is differentiable w.r.t. model parameters when attribution is calculated through Equation 1 and can be trained with most off-the-shelf optimizers. The proposed objective is not dataset-dependent and is applicable to different problem settings such as sentiment classification, abuse detection, etc. It only requires users to specify the target attribution value for tokens of interest in the corpus. We illustrate the effectiveness of our method by applying it to a toxic comment classification problem. In the next section, we first show how we set the target attribution value for identity terms to remove unintended biases while retaining the same performance on the original task. Then, using the same technique, we show how to set target attribution for toxic words to improve classifier performance in a scarce data setting.

Identity	Base	Imp	TOK	Ours
gay	.272	.353	-.006	.000
homosexual	.085	.388	-.006	-.000
queer	.071	.28	-.006	.000
teenage	.030	-0.02	-.006	-.001
lesbian	.012	.046	-.006	.001
vocab avg	-.002	-.001	-.004	-.001

Table 2: Subset of identity terms we used and their mean attribution value on the test set. Method names are abbreviated with the prefix. The last row is the average across all vocabularies.

Template Examples	Labels
I am **Identity**	Non-toxic
Hug **Identity**	Non-toxic
Being **Identity** is wonderful	Non-toxic
I hate **Identity**	Toxic
You are a moronic **Identity**	Toxic
Name is a disgusting **Identity**	Toxic

Table 3: Example templates used to generate the synthetic dataset for evaluation.

4 Experiments

We incorporate human prior in model building on two applications. First, we tackle the problem of unintended bias in toxic comment classification (Dixon et al., 2018) with our proposed method. For our experiments, we aim to mitigate the issue of neutral sentences with identity terms being classified as toxic for a given a set of identity terms. A subset of the identity terms are listed in the first column of Table 2. Second, we force the model to focus on a list of human-selected toxic terms under scarce data scenario to increase model performance.

In the following section, we introduce the dataset we train and evaluate on along with a synthetic dataset to further validate our fairness improvements. After that, we describe our experimental setup. Then, we compare our method to a classifier trained without the prior loss and 2 other baselines. Lastly, we show the results demonstrating usefulness of our approach with data scarcity.

4.1 Dataset

In this work, we use a dataset containing comments from Wikipedia Talk Pages (Dixon et al., 2018). Number of samples are 95,692 / 32,128 / 31,866 in the train/dev/test sets respectively. The ratio of positive (toxic) labels in the training set is 9.7%. The dataset was annotated by human raters, where toxicity was defined as a "rude, disrespectful, or unreasonable comment that is likely to make you leave a discussion" per Dixon et al. (2018). Please refer to the corresponding paper for more details about collection methodology, biases present in the data, and toxicity distribution per comment length.

We also use a synthetically generated dataset to validate our approach on fairness as in Park et al. (2018); Dixon et al. (2018). The dataset is created using a set of templates, developed by Dixon et al. (2018) [2], where placeholders are replaced with different identity terms. We show a subset of example templates in Table 3 and selected identity terms along with their mean attributions across the test set in Table 2. We mainly evaluate the effectiveness of our debiasing technique on this dataset because the original test sets follow the same biased distribution. Intuition is that predictions returned for sentences containing different identity terms in the exact same context should be similar. Hence, this dataset enables us to quantify the performance of a classifier in more detail when controlled on identity.

4.2 Experimental Setup

For the text classifier, we built a convolutional neural network (CNN) classifier as in Kim (2014). The network contains a convolution layer with 128 2-, 3-, 4-gram filters for a sequence length of 100 followed by a max-pooling layer and softmax function. Embeddings were randomly initialized and their size was set to 128. Shorter sequences are padded with <pad> token and longer sequences are truncated. Tokens occurring 5 times or more are retained in the vocabulary. We set dropout as 0.2 and used Adam (Kingma and Ba, 2015) as our optimizer with initial learning rate set to 0.001. We didn't perform extensive network architecture search to improve the performance as it is a reasonably strong classifier with the initial performance of 95.5% accuracy.

The number of interpolating steps for IG is set to 50 (as in the original paper) for calculating Riemann approximation of the integral. Since the output of the binary classification can be reduced to a single scalar output by taking the posterior of the

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

Whole Dataset	Acc	F1	AUC	FP	FN
Baseline	.955	.728	.948	.010	.035
Importance	.957	.739	.953	.009	.034
TOK Replace	.939	.607	.904	.014	.047
Our Method	**.958**	**.752**	**.960**	.009	**.032**
Fine-tuned	.955	.720	.954	**.007**	.038

Table 4: Performance on the Wikipedia toxic comment dataset. Columns represent Accuracy, F-1 score, Area Under ROC curve, False Positive, and False Negative. Numbers represent the mean of 5 runs. Maximum variance is .012.

Identity	Acc	F1	AUC	FP	FN
Baseline	.931	.692	.910	.011	.057
Importance	.933	**.704**	.945	.012	**.055**
TOK Replace	.910	.528	.882	.008	.081
Our Method	**.934**	.697	**.949**	.008	.058
Finetuned	.928	.660	.940	**.007**	.064

Table 5: Performance statistics of all approaches on the Wikipedia dataset filtered on samples including identity terms. Numbers represent the mean of 5 runs. Maximum variance is .001.

positive (toxic) class, the prior is only added to the positive class in equation 3 . We set

$$t_i = \begin{cases} k, & \text{if } x_i \in I \\ a_i, & \text{otherwise} \end{cases}, \quad (5)$$

where I is the set of selected terms and x_i being the i th token in the sequence.

For fairness experiments, we set k to be 0 and I to the set of identity terms with the hope that these terms should be as neutral as possible when making predictions. Hyperparamter λ is searched in the range of $(1, 10^8)$ and increased from 1 by a scale of 10 on the dev set and we pick the one with best F-1 score. λ is set to 10^6 for the final model.

For data scarcity experiments, we set k to 1 and I to the set of toxic terms to force the model to make high attributions to these terms. Hyperparameter λ is set to 10^5 across all data size experiments by tuning on the dev set with model given 1% of training data.

Each experiment was repeated for 5 runs with 10 epochs and the best model is selected according to the dev set. Training takes 1 minute for a model with cross-entropy loss and 30 minutes for a model with *joint* loss on an NVidia V100 GPU. However, reducing the step size in IG for calculating Riemann approximation of the integral to 10 steps reduces the training time to 6 minutes. Lastly, training with *joint* loss reaches its best performance in later epochs than training with cross-entropy loss.

Implementation Decisions

When taking the derivative with respect to the loss, we treat the interpolated embeddings as constants. Thus, the *prior* loss does not back-propagate to the embedding parameters. There are two reasons that lead to this decision: (i) taking the gradient of the interpolate operation would break the axioms that IG guarantees; (ii) the Hessian of the embedding matrix is slow to compute. The implementation decision does not imply that *prior* loss has no effect on the word embeddings, though. During training, the model parameters are updated with respect to both losses. Therefore, the word embeddings had to adjust accordingly to the new model parameters by updating the embedding parameters with cross-entropy loss.

4.3 Results on Incorporating Fairness Priors

We compare our work to 3 models with the same CNN architecture, but different training settings:

- **Baseline:** A baseline classifier trained with cross-entropy loss.
- **Importance:** Classifier trained with cross-entropy loss, but the loss for samples containing identity words are weighted in the range $(1, 10^8)$, where the actual coefficient is determined to be 10 on the dev set based on F-1 score.
- **TOK Replace:** Common technique for making models blind to identity terms (Garg et al., 2018). All identity terms are replaced with a special <**id**> token.

We also explore a different training schedule for cases where a model has been trained to optimize for a classification loss:

- **Finetuned:** An already-trained classifier is finetuned with joint loss for several epochs. The aim of this experiment is to show that our method is also applicable for tweaking trained models, which could be useful if the original had been trained for a long time.

gay			homosexual			<id>
Baseline	Our method	Importance	Baseline	Our method	Importance	Tok Replace
a**hole	<pad>	sh*t	b*tch	scorecard	f*ck	456
f*ck	jus	f*cking	cr*p	dutchman	b*tch	messengers
pathetic	tweaking	b*tch	f*g	‘oh	pu**y	louie
fu*king	sess	f*ck	bulls***	678	sucks	dome
fa**ot	ridiculous	penis	dumba*s	nitrites	f*cked	accumulation
bas**rd	‘do	suck	sh*t	poured	pathetic	ink
cr*p	manhood	pu**y	penis	nuts	c*ck	usher
suck	dub	d*ckhead	moron	gubernatorial	fart	wikiepedia
sh*t	heartening	moron	retard	convincing	a**hole	schizophrenics
a*s	desire	fa**ot	gay	strung	fa**ot	notables

Table 6: Top 10 nearest neighbors for tokens ‘gay’ and ‘homosexual’ and <id> for TOK Replace. All asterisks are inserted by authors to replace certain characters.

Synthetic	AUC	FPED	FNED
Baseline	.885	2.77	3.51
Importance	.850	2.90	3.06
TOK Replace	.930	**0.00**	**0.00**
Our Method	**.952**	0.01	0.11
Finetuned	.925	0.00	0.19

Table 7: AUC and Bias mitigation metrics on synthetic dataset. The lower the better for Bias mitigation metrics and is bounded by 0. Numbers represent the mean of 5 runs. Maximum variance is 0.013.

4.3.1 Evaluation on Original Data

We first verify that the prior loss term does not adversely affect overall classifier performance on the main task using general performance metrics such as accuracy and F-1. Results are shown in Table 4. Unlike previous approaches (Park et al., 2018; Dixon et al., 2018; Madras et al., 2018), our method does not degrade classifier performance (it even improves) in terms of all reported metrics. We also look at samples containing identity terms. Table 5 shows classifier performance metrics for such samples.

The importance weighting approach slightly outperforms the baseline classifier. Replacing identity words with a special tokens, on the other hand, hurts the performance on the main task. One of the reasons might be that replacing all identity terms with a token potentially removes other useful information model can rely on. If we were to make an analogy between the token replacement method and hard ablation, then the same analogy can be made between our method and soft ablation. Hence, the information pertaining to identity terms is not completely lost for our method, but come at a cost.

Results for fine-tuning experiments show the performance after 2 epochs. It is seen that the model converges to similar performance with joint training after only 2 epochs, albeit being slightly poorer.

4.3.2 Evaluation on Synthetic Data

Now we run our experiments on the template-based synthetic data. As stated, this dataset is used to measure biases in the model since it is unbiased towards identities. We use AUC along with False Positive Equality Difference (FPED) and False Negative Equality Difference (FNED), which measure a proxy of Equality of Odds (Hardt et al., 2016), as in Dixon et al. (2018); Park et al. (2018). FPED sums absolute differences between overall false positive rate and false positive rates for each identity term. FNED calculates the same for false negatives. Results on this dataset are shown in Table 7. Our method provides substantial improvement on AUC and almost completely eliminates false positive and false negative inequality across identities.

The fine-tuned model also outperforms the baseline for mitigating the bias. The token replacement method comes out as a good baseline for mitigating the bias since it treats all identities the same. The importance weighting approach fails to produce an unbiased model.

4.4 Nearest Neighbors of Identity Terms

Models convert input tokens to embeddings before providing them to convolutional layers. As embeddings make up the majority of the parameters of the network and can be exported for use in

Ratio	1%		5%		10%	
Toxic	**Base**	**Ours**	**Base**	**Ours**	**Base**	**Ours**
hell	-.002	.035	.002	.673	.076	.624
moron	-.002	.044	.002	.462	.077	.290
sh*t	-.003	.078	.006	.575	.098	.437
f*ck	-.003	.142	.013	.643	.282	.682
b*tch	-.003	.051	.002	.397	.065	.362

Table 8: Subset of toxic terms we used in the experiments and their mean attribution value on the test set for different training sizes.

other tasks, we're interested in how they change for the identity terms. We show 10 nearest neighbors of the terms <id> (for the token replacement method), "gay", and "homosexual" – top two identity terms with the most mean attribution difference (our method vs. baseline), in Table 6.

The word embedding of the term "gay" shifts from having swear words as its neighbors to having the <pad> token as the closest neighbor. Although the term "homosexual" has lower mean attribution, its neighboring words are still mostly swear words in the baseline embedding space. "homosexual" also moved to more neutral terms that shouldn't play a role in deciding if the comment is toxic or not. Although they are not as high quality as one would expect general-purpose word embeddings to be possibly due to data size and the model having a different objective, the results show that our method yields inherently unbiased embeddings. It removes the necessity to initialize word embeddings with pre-debiased embeddings as proposed in Bolukbasi et al. (2016).

The importance weighting technique penalizes the model on the sentence level instead of focusing on the token level. Therefore, the word embedding of "gay" doesn't seem to shift to neutral words. The token replacement method, on the other hand, replaces the identity terms with a token that is surrounded with neutral words in the embedding space, so it results in greater improvement on the synthetic dataset. However, since all identity terms are collapsed into one, it's harder for the model to capture the context and as a result, classification performance on the original dataset drops.

4.5 Results on Incorporating Priors in Different Training Sizes

We now demonstrate our approach on encouraging higher attributions on toxic words to increase model performance in scarce data regime. We down-sample the dataset with different ratios to simulate a data scarcity scenario. To directly validate the effectiveness of prior loss on attributions, we first show that the attribution of the toxic words have higher values for our method across different data ratios compared to the baseline in Table 8. We also show that the attribution for these terms increases as training data increases for the baseline method. We then show model performance on testing data for different data size ratios for the baseline and our method in Figure 1. Our method outperforms the baseline by a big margin in 1% and 5% ratio. However, the impact of our approach diminishes after adding more data, since the model starts to learn to focus on toxic words itself for predicting toxicity without the need for prior injection. We can also see that both the baseline and our method start to catch up with the rule based approach, where we give positive prediction if the toxic word is in the sentence, and eventually outperform it.

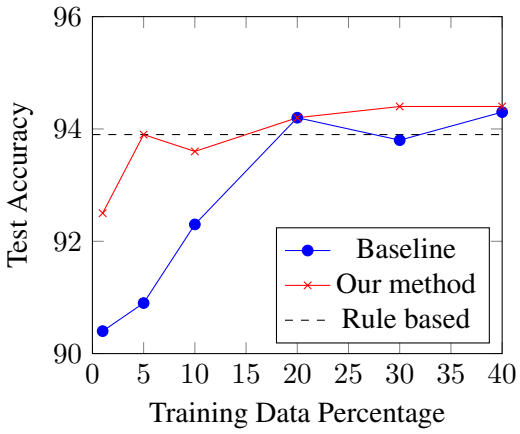

Figure 1: Test accuracy for different training sizes. The rule based method gives positive prediction if the comment includes any of the toxic temrs.

5 Discussion and Related Work

For explaining ML models, recent research attempts offer techniques ranging from building inherently interpretable models (Kim et al., 2014) to building a proxy model for explaining a more complex model (Ribeiro et al., 2016; Frosst and Hinton, 2017) to explaining inner mechanics of mostly uninterpretable neural networks (Sundararajan et al., 2017; Bach et al., 2015). One family of interpretability methods uses sensitivity of the network with respect to data points (Koh

and Liang, 2017) or features (Ribeiro et al., 2016) as a form of explanation. These methods rely on small, local perturbations and check how a network's response changes. Explaining text models has another layer of complexity due to a lock of proper technique to generate counterfactuals in the form of small perturbations. Hence, interpretability methods tailored for text are quite sparse (Mudrakarta et al., 2018; Jia and Liang, 2017; Murdoch et al., 2018).

On the other hand, there are many papers criticizing the aforementioned methods by questioning their faithfulness, correctness (Adebayo et al., 2018; Kindermans et al., 2017) and usefulness. Smilkov et al. (2017) show that gradient based methods are susceptible to saturation and can be fooled by adversarial techniques. Other sets of papers (Miller, 2019; Gilpin et al., 2018) attack model explanation papers from a philosophical perspective. However, the lack of actionability angle is often overlooked. Lipton (2018) briefly questions the practical benefit of having model explanations from a practitioners perspective. There are several works taking advantage of model explanations. Namely, using model explanations to aid doctors in diagnosing retinopathy patients (Sayres et al., 2018), and removing minimal features, called pathologies, from neural networks by tuning the model to have high entropy on pathologies (Feng et al., 2018). The authors of Ross et al. (2017) propose a similar idea to our approach in that they regularize input gradients to alter the decision boundary of the model to make it more consistent with domain knowledge. However, the input gradients technique has been shown to be an inaccurate explanation technique (Adebayo et al., 2018).

Addressing and mitigating bias in NLP models are paramount tasks as the effects on these models adversely affect protected subpopulations (Schmidt and Wiegand, 2017). One of the earliest works is Calders and Verwer (2010). Later, Bolukbasi et al. (2016) proposed to unbias word vectors from gender stereotypes. Park et al. (2018) also try to address gender bias for abusive language detection models by debiasing word vectors, augmenting more data and changing model architecture. While their results seem to show promise for removing gender bias, their method doesn't scale for other identity dimensions such as race and religion. The authors of Dixon et al. (2018) highlight the bias in toxic comment classifier models originating from the dataset. They also supplement the training dataset from Wikipedia articles to shift positive class imbalance for sentences containing identity terms to dataset average. Similarly, their approach alleviates the issue to a certain extent, but does not scale to similar problems as their augmentation technique is too data-specific. Also, both methods trade original task accuracy for fairness, while our method does not. Lastly, there are several works (Davidson et al., 2017; Zhang et al., 2018b) offering methodologies or datasets to evaluate models for unintended bias, but they fail to offer a general framework.

One of the main reasons our approach improves the model in the original task is that the model is now more robust thanks to the reinforcement provided to the model builder through attributions. From a fairness angle, our technique shares similarities with adversarial training (Zhang et al., 2018a; Madras et al., 2018) in asking the model to optimize for an additional objective that transitively unbiases the classifier. However, those approaches work to remove protected attributes from the representation layer, which is unstable. Our approach, on the other hand, works with basic human-interpretable units of information – tokens. Also, those approaches propose to sacrifice main task performance for fairness as well.

While our method enables model builders to inject priors to aid a model, it has several limitations. In solving the fairness problem in question, it causes the classifier to not focus on the identity terms even for the cases where an identity term itself is being used as an insult. Moreover, our approach requires prior terms to be manually provided, which bears resemblance to blacklist approaches and suffers from the same drawbacks. Lastly, the evaluation methodology that we and previous papers (Dixon et al., 2018; Park et al., 2018) rely on are based on a synthetically-generated dataset, which may contain biases of the individuals creating it.

6 Conclusion and Future Work

In this paper, we proposed actionability on model explanations that enable ML practitioners to enforce priors on their model. We apply this technique to model fairness in toxic comment classification. Our method incorporates Path Integrated Gradients attributions into the objective function

with the aim of stopping the classifier from carrying along false positive bias from the data by punishing it when it focuses on identity words.

Our experiments indicate that the models trained jointly with cross-entropy and prior loss do not suffer a performance drop on the original task, while achieving a better performance in fairness metrics on the template-based dataset. Applying model attribution as a fine-tuning step on a trained classifier makes it converge to a more *debiased* classifier in just a few epochs. Additionally, we show that model can be also forced to focus on pre-determined tokens.

There are several avenues we can explore as future research. Our technique can be applied to implement a more robust model by penalizing the attributions falling outside of tokens annotated to be relevant to the predicted class. Another avenue is to incorporate different model attribution strategies such as DeepLRP (Bach et al., 2015) into the objective function. Finally, it would be worthwhile to invest in a technique to extract problematic terms from the model automatically rather than providing prescribed identity or toxic terms.

Acknowledgments

We thank Salem Haykal, Ankur Taly, Diego Garcia-Olano, Raz Mathias, and Mukund Sundararajan for their valuable feedback and insightful discussions.

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian J. Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. In *Proceedings of NeurIPS*.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine bias: Theres software used across the country to predict future criminals. and its biased against blacks. *ProPublica*.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, Wojciech Samek, and Oscar Deniz Suarez. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. In *Proceedings of PloS one*.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of NIPS*.

Toon Calders and Sicco Verwer. 2010. Three naive bayes approaches for discrimination-free classification. In *Proceedings of Data Mining and Knowledge Discovery*, Hingham, MA, USA. Kluwer Academic Publishers.

Irene Chen, Fredrik D. Johansson, and David Sontag. 2018. Why is my classifier discriminatory? In *Proceedings of NeurIPS*.

Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of ICWSM*.

Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Pai-Shun Ting, Karthikeyan Shanmugam, and Payel Das. 2018. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Proceedings of NeurIPS*.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of AIES*.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of EMNLP*.

Nicholas Frosst and Geoffrey E. Hinton. 2017. Distilling a neural network into a soft decision tree. *Arxiv, 1711.09784*.

Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed Huai hsin Chi, and Alex Beutel. 2018. Counterfactual fairness in text classification through robustness. In *Proceedings of AIES*.

Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *Proceedings of DSAA*.

Bryce Goodman and Seth Flaxman. 2016. European union regulations on algorithmic decision-making and a right to explanation. In *Proceedings of ICML Workshop on Human Interpretability in Machine Learning*.

Moritz Hardt, Eric Price, and Nathan Srebro. 2016. Equality of opportunity in supervised learning. In *Proceedings of NIPS*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of EMNLP*.

Been Kim, Cynthia Rudin, and Julie Shah. 2014. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *Proceedings of NIPS*.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda B. Viégas, and Rory Sayres. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Proceedings of ICML*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*.

Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. 2017. The (un)reliability of saliency methods. In *Proceedings of NIPS workshop on Explaining and Visualizing Deep Learning*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of ICML*.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of EMNLP*.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in nlp. In *Proceedings of NAACL-HLT*.

Zachary C. Lipton. 2018. The mythos of model interpretability. In *Queue*, New York, NY, USA. ACM.

Yin Lou, Rich Caruana, and Johannes Gehrke. 2012. Intelligible models for classification and regression. In *Proceedings of KDD*.

Scott M. Lundberg, Gabriel G. Erion, and Su-In Lee. 2018. Consistent individualized feature attribution for tree ensembles. In *Proceedings of KDD*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of NIPS*.

David Madras, Elliot Creager, Toniann Pitassi, and Richard S. Zemel. 2018. Learning adversarially fair and transferable representations. In *Proceedings of ICML*.

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. In *Proceedings of Artificial Intelligence*.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did the model understand the question? In *Proceedings of ACL*.

W. James Murdoch, Peter J. Liu, and Bin Yu. 2018. Beyond word importance: Contextual decomposition to extract interactions from lstms. In *Proceedings of ICLR*.

Ji Ho Park, Jamin Shin, and Pascale Fung. 2018. Reducing gender bias in abusive language detection. In *Proceedings of EMNLP*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of KDD*.

Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. 2017. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of IJCAI*, IJCAI'17, pages 2662–2670. AAAI Press.

Rory Sayres, Ankur Taly, Ehsan Rahimy, Katy Blumer, David Coz, Naama Hammel, Jonathan Krause, Arunachalam Narayanaswamy, Zahra Rastegar, Derek Wu, Shawn Xu, Scott Barb, Anthony Joseph, Michael Shumski, Jesse Smith, Arjun B. Sood, Greg S. Corrado, Lily Peng, and Dale R. Webster. 2018. Using a deep learning algorithm and integrated gradients explanation to assist grading for diabetic retinopathy. In *Proceedings of American Academy of Ophthalmology*.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of International Workshop on Natural Language Processing for Social Media*.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of ICML*.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise. In *Proceedings of ICML Workshop on Visualization for Deep Learning*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of ICML*.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL Student Research Workshop*.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of WWW*.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018a. Mitigating unwanted biases with adversarial learning. In *Proceedings of AIES*.

Ziqi Zhang, David Robinson, and Jonathan A. Tepper. 2018b. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *Proceedings of ESWC*.